ENCICLOPEDIA PRACTICA DE LA

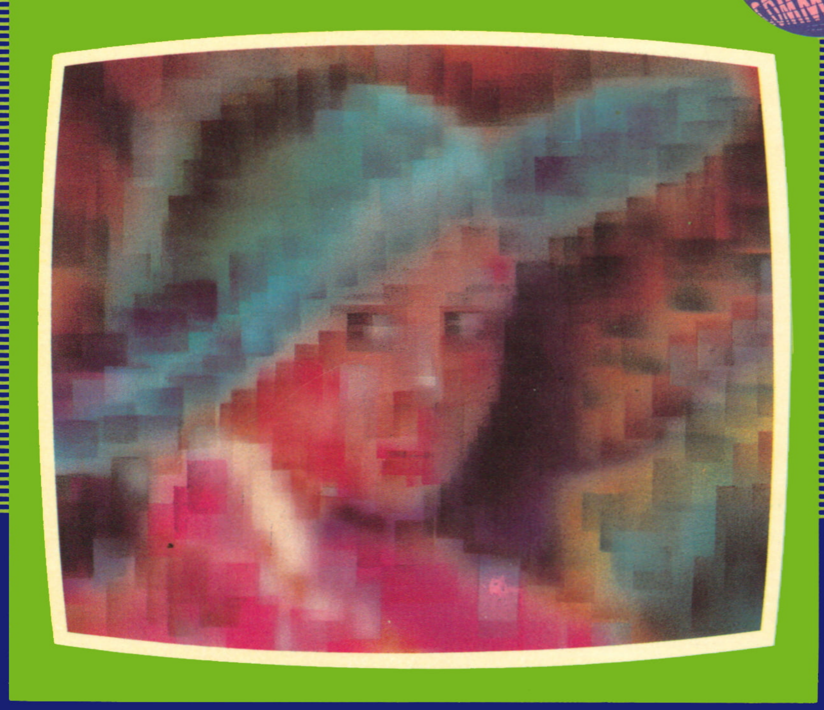
INFORMATICA APLICADA

20

Visión artificial: tratamiento de imágenes por computador

S. Figueroa, G. Cristóbal y J. I. Couto

SINCLAIR
AMSTRAD
COMMODORE
IBM



EDICIONES SIGLO CULTURAL

ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

20

Visión artificial: tratamiento de imágenes por computador

EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S.A.

Director-editor:
RICARDO ESPAÑOL CRESPO.

Gerente:
ANTONIO G. CUERPO.

Directora de producción:
MARIA LUISA SUAREZ PEREZ.

Directores de la colección:
MANUEL ALFONSECA, Doctor Ingeniero de Telecomunicación
y Licenciado en Informática
JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño y maquetación:
BRAVO-LOFISH.

Dibujos:
JOSE OCHOA Y ANTONIO PERERA.

Tomo XX. **Visión artificial: tratamiento de imágenes por computador**

S. FIGUEROA, Ingeniero Técnico de Electrónica
G. CRISTOBAL, Doctor Ingeniero de Telecomunicación
J. I. COUTO, Ingeniero Electrónico

Ediciones Siglo Cultural, S.A.

Dirección, redacción y administración:
Sor Angela de la Cruz, 24-7.º G. Teléf. 279 40 36. 28020 Madrid.

Publicidad:
Gofar Publicidad, S.A. San Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:
COEDIS, S.A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.
Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:
CADE, S.R.L. Pasaje Sud América. 1532. Teléf.: 21 24 64.
Buenos Aires - 1.290. Argentina.

ISBN del tomo: 84-7688-063-4
ISBN de la obra: 84-7688-018-9.

Fotocomposición:
ARTECOMP, S.A. Albarracín, 50. 28037 Madrid.

Imprime:
MATEU CROMO. Pinto (Madrid).

Depósito legal: M. 5.046-1987
Printed in Spain - Impreso en España.

Suscripciones y números atrasados:
Ediciones Siglo Cultural, S.A.
Sor Angela de la Cruz, 24-7.° G. Teléf. 279 40 36. 28020 Madrid

Abril 1987
P.V.P. Canarias: 365,-

INDICE

INTRODUCCION 1

ANTECEDENTES HISTORICOS



EL interés por el procesado digital de imágenes se divide en dos áreas principales de aplicación: mejora de imágenes (para su mejor interpretación por el hombre) y procesado de imágenes (para su interpretación automática por medio de una máquina). Las primeras técnicas de tratamiento de imágenes se remontan al principio de los años veinte, cuando se trataba de mejorar las fotografías de periódicos enviadas por cable submarino entre Londres y Nueva York. El uso de imágenes en forma digital se remonta a 1921, con la introducción del primer sistema de transmisión de fotografías por cable submarino desde Londres a Nueva York. La introducción del sistema Barlane de transmisión fotográfica por cable redujo el tiempo empleado en transportar una fotografía a través del Atlántico de una semana a menos de tres horas. Los primeros sistemas Barlane eran capaces de codificar imágenes en cinco niveles distintos de intensidad y esta capacidad fue aumentada en 1929 a quince niveles. Durante este período también se mejoró notablemente el proceso de reproducción, que pasó de ser la impresora telegráfica a un sistema con placa fotográfica impresionada con rayos de luz modulados por la imagen codificada.

Aunque continuaron las mejoras en los métodos de procesado durante los siguientes treinta y cinco años, la gran potencia de cálculo de los computadores, derivada de la progresiva evolución de las tecnologías VLSI y el programa espacial fueron los que dieron gran impulso a los conceptos de procesado de imagen. El tratamiento digital de imágenes entendido como la aplicación de procedimientos matemáticos tendentes a corregir las fotografías y mejorar su interpretabilidad no se inició hasta 1963, cuando en los laboratorios de investigación de IBM se empezó a procesar mediante ordenador las imágenes obtenidas por el satélite *Tiros*. A partir de ese momento, el tratamiento digital de imágenes se ha convertido en un

área de las ciencias aplicadas que viene experimentando un crecimiento acelerado durante los últimos veinte años. El uso de técnicas de mejora de imagen por ordenador comenzó en el Jet Propulsion Laboratory (Pasadena, California) en 1964, cuando fotografías de la Luna transmitidas por el *Ranger 7* eran procesadas por un computador para corregir varios tipos de distorsión de la imagen inherente a la cámara de TV de a bordo. Estas técnicas sirvieron como base para posteriores métodos usados en la mejora y restauración de imágenes para programas tan familiares como las misiones Surveyor a la Luna, la serie de misiones Mariner a Marte y los vuelos tripulados Apollo a la Luna. Este desarrollo, inicialmente, se efectuó con objeto de tratar las imágenes suministradas por las diversas misiones espaciales, en especial las de los satélites dedicados a la teledetección de recursos naturales e impulsada por la utilización de computadores cada vez más potentes, tuvo menor incidencia en otras áreas, si bien en algunos casos, como el de la microscopía electrónica, estas técnicas se han empleado desde mediados de la década de los sesenta.

Desde 1964 hasta ahora, el campo del procesado de imágenes ha experimentado un vigoroso crecimiento. Además del programa espacial, las técnicas más usadas hoy se dedican a una gran variedad de campos como: medicina (imágenes biomédicas y rayos X), geografía y meteorología (análisis de imágenes aéreas y de satélite), arqueología, astronomía, biología, medicina nuclear, defensa o industria. Todas ellas no sólo en cuanto a la mejora de imágenes para interpretación humana, sino también en lo referente a su interpretación por una máquina en campos como: reconocimiento de caracteres, robots industriales para montaje e inspección, reconocimiento de objetivos militares, procesado de huellas dactilares, análisis de muestras de sangre y radiográficas, predicción del tiempo atmosférico y de las cosechas, acústica, comunicaciones, geofísica, procesamiento de voz, microscopía, teledetección, etc.

Por otra parte, se ha producido un gran acercamiento entre las técnicas de procesado digital de imágenes y la de gráficos por ordenador *(computer graphics)*, hasta el punto de que ambas se solapan en el marco de ciertas aplicaciones.

El tratamiento digital de imágenes abarca una gran variedad de técnicas y herramientas matemáticas que se han desarrollado, si bien cabe señalar que en los últimos años se han comenzado a desarrollar bases científicas en lo que antes era un simple conjunto de técnicas.

El tratamiento digital de imágenes abarca una gran variedad de técnicas y herramientas matemáticas que pueden ser clasificadas en los siguientes puntos:

1. Digitalización y cuantificación, que consiste en efectuar el paso de imágenes continuas a discretas.

2. Operaciones de preprocesado, consistentes en la realización de determinadas operaciones tendentes a normalizar los datos de entrada.

3. Mejora o realce de imágenes, consistente en efectuar algunas operaciones tendentes a mejorar la apariencia de los datos de entrada.

4. Restauración de imágenes, consistente en la recuperación de la imagen original, en el supuesto de que los datos de entrada se encuentran afectados de algún tipo de degradación, como, por ejemplo, desenfoque, turbulencia atmosférica, movimiento, etc.

5. Codificación de imágenes, consistente en la compresión de la información con objeto de reducir los costes de transmisión de las imágenes, así como conseguir un almacenamiento más eficiente, como sucede, por ejemplo, en el caso de las imágenes médicas, en ingeniería, etc.

6. Análisis de imágenes, consistente en la obtención de descripciones numéricas o simbólicas de la imagen de partida. Las descripciones de una imagen se refieren generalmente a las partes constituyentes de dicha imagen. Es por ello que, el primer paso en todo proceso de análisis de imágenes lo constituye la denominada segmentación en regiones con significado propio.

7. Reconstrucción de imágenes, consistente en la obtención de información acerca de un objeto a partir de sus proyecciones, como, por ejemplo, la realizada en tomografía, en donde se obtienen secciones transversales del cuerpo humano, o los métodos basados en la resonancia magnética nuclear.

Estas técnicas adoptan en la práctica formas y modos de operación muy diversos en función del tipo de imágenes a que se aplican y del tipo de información que de ellas se desea extraer.

REPRESENTACION DIGITAL DE IMAGENES

Antes de comenzar con la descripción de algunas de las técnicas del procesado digital de imágenes, es preciso definir qué se entiende por una imagen digital, a fin de poder apreciar las posibilidades de estas técnicas, así como comprender la problemática que entraña su utilización.

En el ámbito del tratamiento de imágenes, el término *imagen* se refiere a una función bidimensional de la intensidad luminosa f(x,y), donde x e y denotan coordenadas espaciales y el valor de f en cada punto (x,y) es pro-

porcional al brillo, intensidad o nivel de gris de la imagen en ese punto. A veces es útil visualizar una función imagen en perspectiva tridimensional, siendo la intensidad el tercer eje.

Una *imagen digital* f (x,y) se obtiene a partir de la radiancia de un objeto que ha sido discretizada en ambas coordenadas espaciales y en intensidad. Podemos considerar una imagen digital como una *matriz* cuyos índices de fila y columna identifican un punto de la imagen y el correspondiente elemento de la matriz representa el nivel de gris de ese punto. Los elementos de dicha matriz digital se llaman elementos de imagen o *pixels* (o pels), como abreviatura de «picture elements».

Aunque el tamaño de una imagen digital varía con la aplicación, hay grandes ventajas en elegir matrices cuadradas cuyos tamaños y número de niveles de gris sean potencias enteras de 2. Por ejemplo, un tamaño típico comparable en calidad a una imagen de TV monocromática es el de una matriz de 512 × 512 con 128 niveles de gris.

Una de las características que define el procesado de imágenes es el elevado volumen de información que conlleva este tipo de procesos. Este hecho hace que el coste computacional de las operaciones que se efectúen con imágenes sea muy elevado, a menos que éstas sean muy sencillas.

ELEMENTOS DE UN SISTEMA DE PROCESADO DIGITAL DE IMAGENES

Los elementos de un sistema de procesado de imagen básico y de propósito general son los de la figura 1.

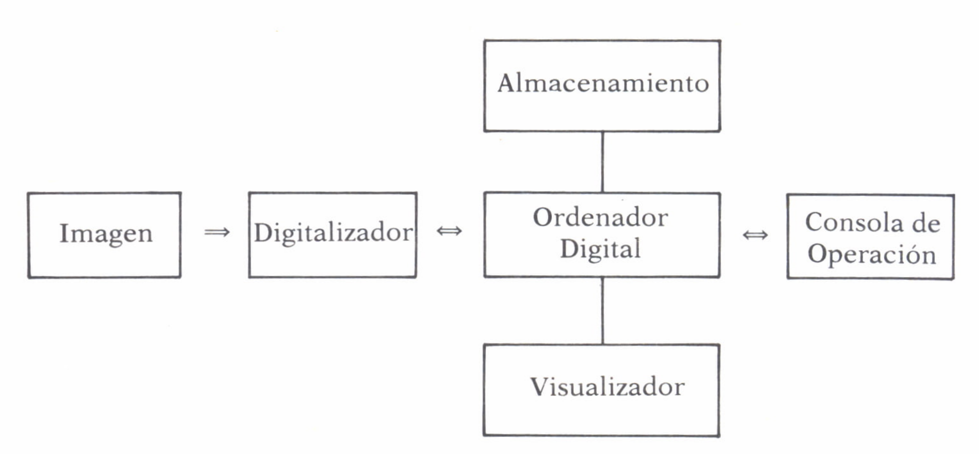


Fig. 1. Elementos de un sistema de procesamiento digital de imágenes.

La operación de un sistema de este tipo puede dividirse en tres categorías principales: digitalización, procesado y visualización.

a) *Digitalizadores*

Convierten una imagen en su representación numérica, de forma que pueda ser introducida en un ordenador digital. Los dispositivos digitalizadores más comunes son:

— **Microdensitómetros.** En ellos la transparencia o fotografía se monta en una plataforma sobre la que se lanza un fino haz de luz. La plataforma se hace girar o mover y se van obteniendo los valores puntuales. El rayo reflejado (o el que la atraviesa, en el caso de transparencias) es enfocado hacia un fotodetector y el nivel de gris es codificado en base a la intensidad del rayo en cada punto. Una imagen digital se obtiene permitiendo sólo valores discretos de intensidad y posición a la salida. Son dispositivos lentos pero de una gran precisión.

— **Scanners «Flying Spot».** También operan sobre el principio de enfocar un haz de luz reflejado o transmitido sobre un foto-detector. En este caso, sin embargo, la imagen permanece estática y la fuente luminosa es un tubo de rayos catódicos (CRT) en el cual un haz de electrones incide sobre una superficie fluorescente. El rayo produce un punto de luz que se mueve y analiza la imagen sobre la superficie del tubo. Ya que el rayo se mueve electrónicamente, permite grandes velocidades de lectura. También permite gran flexibilidad a la hora de leer una imagen, ya que la posición del rayo puede ser rápida y fácilmente establecida externamente.

— **Disectores de imagen.** Su operación se basa en el principio de emisión electrónica.

— **Cámaras de TV.** Muchos digitalizadores de cámaras de TV emplean un tubo vidicón, cuya operación se basa en el principio de la fotoconductividad. Una imagen enfocada en la superficie del tubo produce un patrón de variación de conductividad que corresponde a la distribución de intensidad de la imagen óptica. Estos dispositivos son menos precisos que los anteriores, pero generalmente más baratos.

b) *Procesadores de imagen*

Los dispositivos usados para el procesado de imagen varían desde microprocesadores para aplicaciones de propósito general hasta enormes sistemas de computadores capaces de realizar gran variedad de funciones en imágenes matriciales de alta resolución (computadores de propósito especial).

c) *Visualizadores*

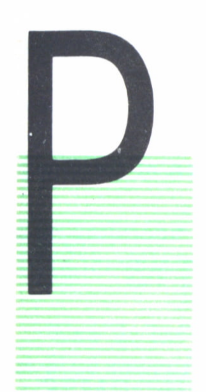
Son los dispositivos que se encargan de convertir las matrices numéricas almacenadas en el ordenador en una representación adecuada para la interpretación humana. Hay tres tipos principales de dispositivos visualizadores:

— **C.R.T.** Los tubos de rayos catódicos (C.R.T.) son sistemas en los que la posición horizontal y vertical de cada elemento de la matriz imagen es convertida en voltajes que se usan para dirigir un haz de electrones, de forma que se produce la imagen de salida bidimensional. En cada punto de la pantalla, la intensidad del haz es modulada por un voltaje proporcional al valor del punto correspondiente en la matriz (pixel).

— **Monitores de TV.** Estos dispositivos convierten la imagen almacenada en el ordenador en una señal de vídeo que puede ser visualizada. Su ventaja es que pueden ofrecer una gama de tonalidades que se aproxima en calidad a la de una placa fotográfica, produciendo una salida que puede ser fácilmente asimilada por el sistema visual. Su inconveniente es que el monitor debe tener un ciclo de refresco de la imagen con una frecuencia de unos 30 «frames» (marcos o imágenes) por segundo para evitar el parpadeo, mientras que la mayoría de los ordenadores convencionales no son capaces de transferir datos a esta velocidad. Para solucionar esto, se dota al sistema de TV de un medio de almacenamiento o «buffer» para transferir datos al monitor a la frecuencia de vídeo.

— **Impresoras.** Los dispositivos de impresión de imagen son útiles para trabajos de procesado de imágenes a baja resolución. Un medio de generar los diferentes niveles de gris de la imagen directamente en el papel es aprovechando la capacidad de «over-strike» (sobre-impresión) de las impresoras convencionales. El nivel de gris de cada pixel puede ser controlado por el número y la intensidad de los caracteres sobreimpresionados en él. Seleccionando adecuadamente el conjunto de caracteres, puede conseguirse una distribución de grises aceptable con un programa sencillo y relativamente pocos caracteres. Este sistema de sobreimpresión de caracteres es el que se ha empleado para visualizar las imágenes definidas en los programas del apéndice 2, proporcionando un grado de resolución suficiente para el propósito de entrenamiento al que se hallan dirigidos.

OPERADORES SOBRE IMAGENES 2



MUESTREO Y CUANTIFICACION DE IMAGENES

PARA poder procesar una imagen f(x,y), esta función debe pasar por un proceso previo de digitalización tanto espacial como en amplitud. La digitalización de las coordenadas espaciales es lo que se llama muestreo de la imagen y la digitalización en amplitud (cuantificación en niveles de grises).

Una imagen continua la podemos aproximar por una matriz de dimensiones N × N, de forma que:

$$f(x,y)= \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,N-1) \end{bmatrix}$$

En esta matriz, cada elemento f(x,y) es un valor discreto. Una matriz de este tipo se denomina imagen digital, y cada uno de los elementos recibe el nombre de pixel.

Para realizar el proceso de digitalización, hay que fijar previamente el valor de N, así como el número de niveles de grises que se van a permitir para cada pixel. Normalmente, se eligen para estos valores potencias enteras de dos, de manera que: $N=2^n$ y $G=2^m$, siendo G el número de niveles de gris. Según esto, el número de bits necesario para almacenar una imagen digitalizada será de N × N pixels de dimensión y m bits/pixel será:

$$b = N \times N \times m$$

Así, por ejemplo, para almacenar una imagen de 128 × 128 pixels y de 64 niveles de gris se necesitarán 98.304 bits.

Un factor importante a tener en cuenta es la determinación de cuántos niveles de gris y qué valor de N se necesitan para obtener una buena aproximación de la imagen. La resolución depende mucho de los valores que se den a m y a N, de forma que cuanto más altos sean éstos, mayor será la resolución de la imagen. Sin embargo, los requerimientos de memoria y

proceso aumentan considerablemente al aumentar estos parámetros. Para obtener una imagen de calidad análoga a la de TV se necesitan 512 × 512 pixels y 128 niveles de grises. Como norma general, un sistema que procese imágenes debe ser capaz de mostrar imágenes de 256 × 256 pixels y 64 niveles de grises, como mínimo.

El efecto que produce la reducción de N es una imagen digital, es la aparición de un efecto damero en la misma. Al disminuir m, el efecto que se produce es la aparición de falsos contornos. Estos son los efectos que produce la variación de N y m por separado; en 1965 Huang estudió las relaciones entre los dos parámetros, haciéndolos variar a un tiempo. Para este experimento utilizó tres imágenes con distintos niveles de detalle; a las conclusiones a que llegó fueron:

1. Al aumentar N y m aumenta la calidad de imagen.
2. En algunos casos, para un N fijo se mejora la imagen al disminuir m, esto es debido a que se aumenta el contraste.
3. En imágenes con una gran cantidad de detalle, se necesitan sólo unos pocos niveles de grises para conseguir una buena aproximación.



MUESTREO Y CUANTIFICACION NO UNIFORMES

Fijado el valor de N, en muchos casos es posible mejorar el aspecto de la imagen usando un método adaptativo, en el que el proceso de muestreo depende de las características de la imagen. En general, se necesita un mayor número de muestras en aquellas regiones en que las transiciones entre distintos niveles de grises son más pronunciadas, haciéndose, por tanto, un muestreo más preciso, pudiéndose tomar un número menor de muestras en las regiones de la imagen donde las transiciones son más suaves.

Vamos a considerar, por ejemplo, la imagen de un rostro sobre un fondo uniforme. El fondo queda representado adecuadamente si tomamos un número pequeño de muestras, mientras que no ocurre lo mismo con el rostro, por lo que hay que aumentar el número de muestras tomadas de este último, mejorándose así la imagen.

Se ve claramente que hay imágenes en las que distribuir el número de muestras de forma no uniforme resulta ventajoso. Una buena técnica consiste en tomar un número mayor de muestras en las fronteras de transición entre niveles de grises.

El muestreo no uniforme se usa para resaltar los bordes; es por esto que no debe usarse en imágenes con regiones uniformes relativamente pequeñas, por ejemplo, la imagen de una masa de gente apiñada. También se usa esta técnica cuando se quiere obtener una imagen más detallada.

Se usa una técnica parecida a la expuesta anteriormente en el muestreo no uniforme, pero jugando aquí con la distribución de los niveles de grises.

Hay varias formas de realizar una cuantificación no uniforme sobre una imagen. Una de ellas consiste en utilizar pocos niveles en los bordes, de manera que se consiga resaltarlos, y tomar más niveles en los cambios suaves, consiguiendo la reducción y en algunos casos la desaparición de falsos contornos.

Otra alternativa consiste en realizar una mayor cuantificación en los niveles de grises que sean más frecuentes, haciendo una cuantificación menor en aquellos niveles que sean menos frecuentes.

Al igual que el muestreo no uniforme, la cuantificación no uniforme se usa para resaltar bordes y obtener imágenes con mayor detalle de representación.



LA TRANSFORMADA DE FOURIER

Uno de los problemas que plantea el estudio de los fenómenos físicos es el de su representación. Dicha representación requiere la incorporación de conceptos que son idealizaciones introducidas por el observador como modelización de la realidad. Las representaciones más habituales de una señal son la temporal (o espacial) y la de frecuencias temporales (o espaciales). Esta última transformada de Fourier de la primera, permite acceder a las componentes espectrales de la señal (representación armónica). Ambos tipos de representación son completas y totalmente equivalentes.

Como es bien conocido, una función periódica puede ser descompuesta mediante suma de infinitos senos y cosenos con diferentes amplitudes, frecuencias y fases. En el caso de funciones no periódicas, la serie infinita se convierte en una integral de la forma:

$$F(u) = \int_{-\infty}^{+\infty} f(x) \exp(-j2\pi ux)\, dx \qquad (3.1)$$

en donde $\exp(-j2\pi ux) = \cos(2\pi ux) - j\mathrm{sen}(2\pi ux)$, siendo $j = -1$ la unidad imaginaria.

La Ec. (3.1) proporciona la transformada de Fourier unidimensional F(u) de la señal f(x). Si la función f(x) es real, su correspondiente transformada de Fourier es compleja, es decir:

$$F(u) = R(u) + jI(u) \qquad (3.2)$$

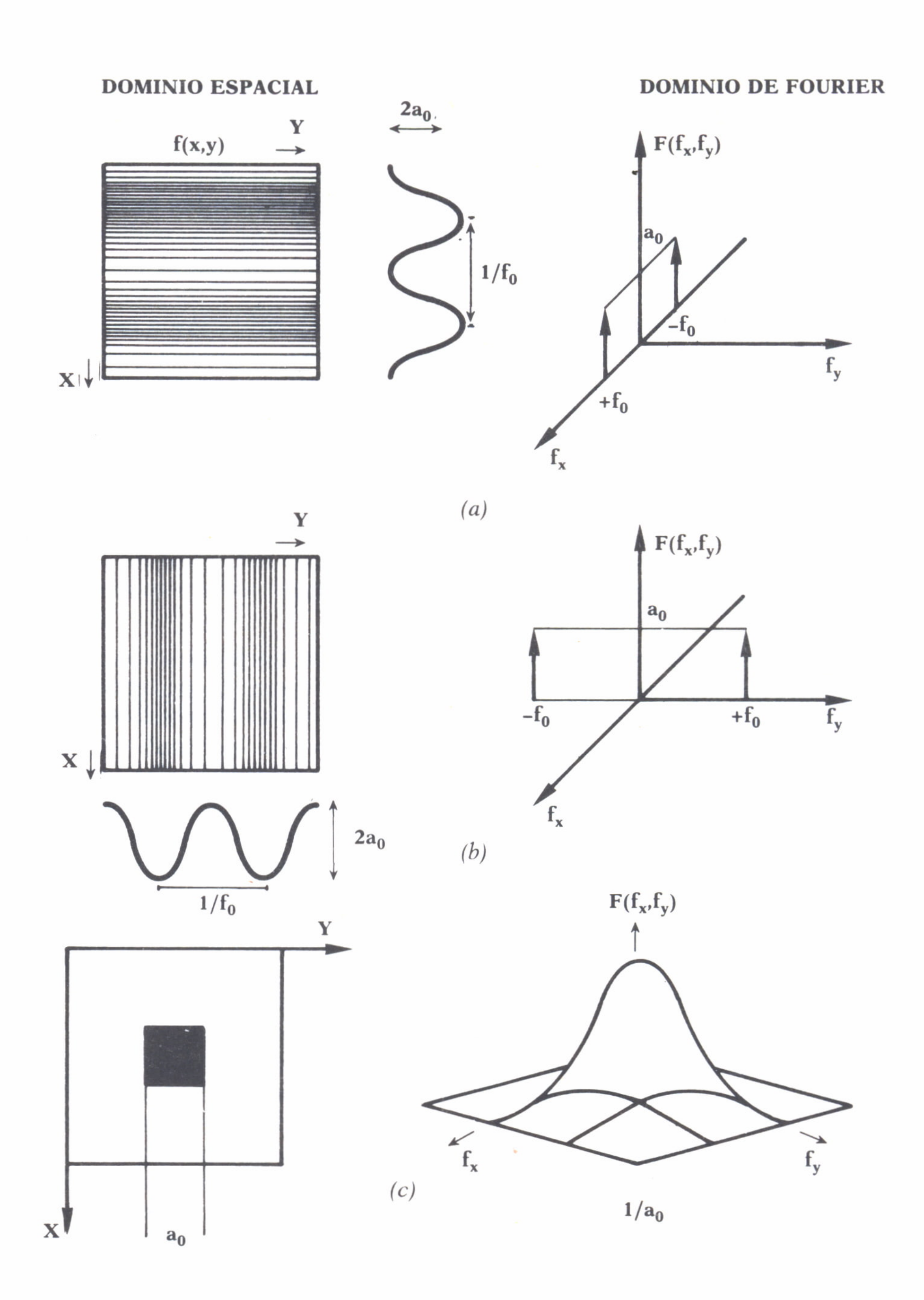


Fig. 1. Representación en el dominio espacial y de frecuencia de funciones sinusoidales (a,b), bidimensionales y de una función rectángulo bidimensional (c).

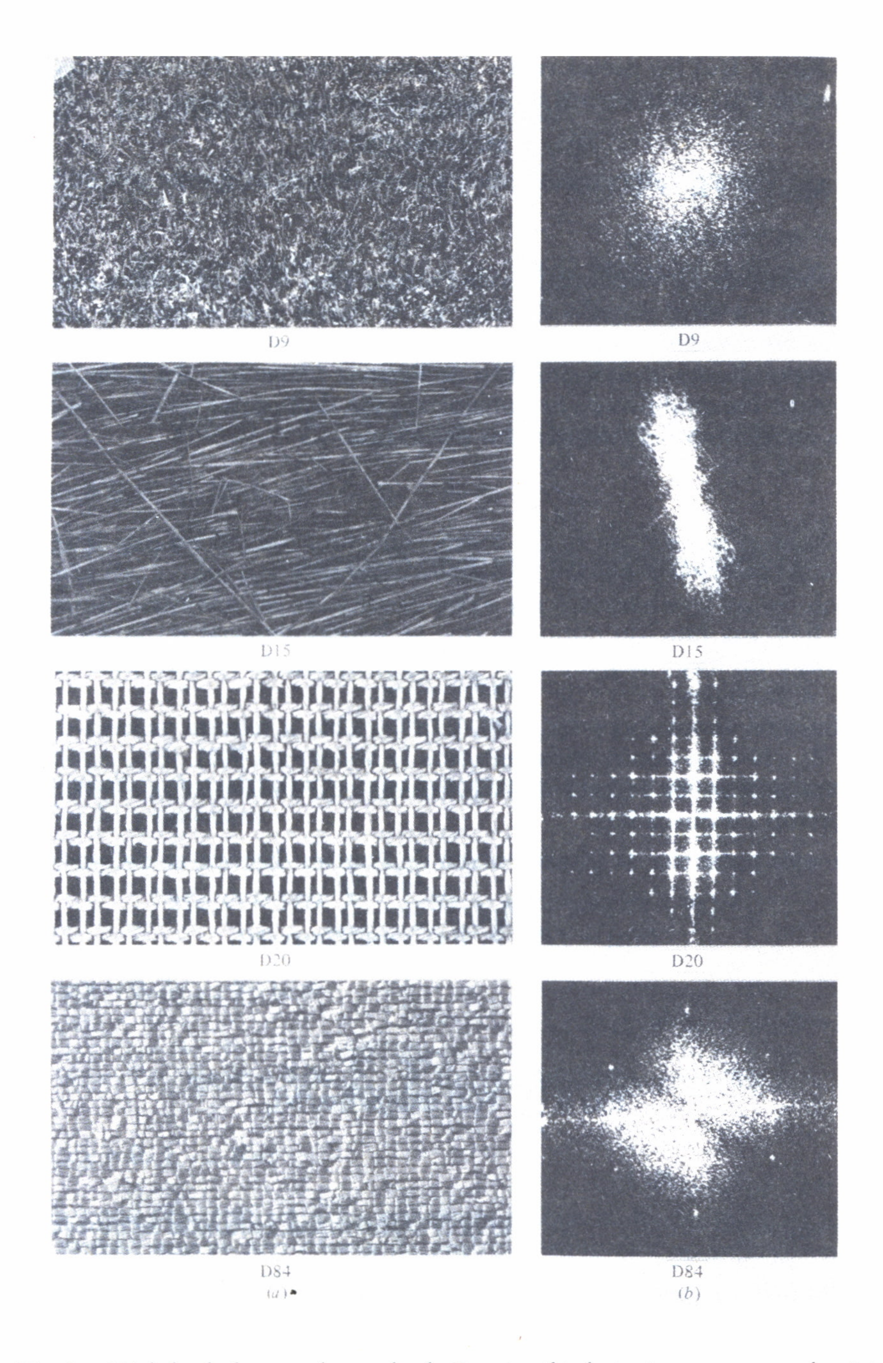


Fig. 2. Módulo de la transformada de Fourier (b) de imágenes texturadas (a).

siendo R(u) e I(u) las partes real e imaginaria de F(u). También puede representarse F(u) en la forma módulo-argumental, es decir:

$$F(u) = M(u) \exp(j\Phi(u)) \qquad (3.3)$$

siendo $M(u) = \sqrt{R^2 + I^2}$

y $\Phi(u) = tg^{-1}(I(u)/R(u))$

Al módulo de la transformada de Fourier F(u) se le denomina espectro de Fourier y a Φ(u) ángulo de fase. En la figura 1 se presentan algunos ejemplos de señales con variación sinusoidal y rectangular (izquierda) y su correspondiente transformación de Fourier (derecha). En la figura 2 se representan los módulos de la transformación de Fourier de algunos ejemplos de imágenes con información de textura.



Transformada de Fourier discreta

Sea f(x) una función continua que se ha discretizado en la secuencia $\{f(x_0), f(x_0 + \delta), f(x_0 + 2\delta), ..., f(x_0 + [N-1]\delta)\}$, tomando N muestras. separadas por δ unidades.

Definimos ahora $f(x) = f(x_0+x\delta)$, y la variable x tomará valores discretos 0,1,2,...,N-1, es decir, consideramos la secuencia $\{f(0), f(1), ..., f(N-1)\}$ para representar los valores de N muestras equidistanciadas, tomadas de una función continua f(x).

Con esta notación, la transformada de Fourier discreta se define como:

$$F(u) = 1/N \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N],$$

para u = 0, 1, 2, ..., N–1, y:

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[j2\pi ux/N],$$

para x = 0, 1, 2, ..., N–1

Los valores u = 0, 1, ..., N–1 de la transformada continua se corresponden con las muestras de la discreta para 0, δ, 2δ, ..., (N–1)δ. Por otro lado, la relación que liga δ(x) con δ(u) viene dada por la siguiente expresión:

$$\delta(u) = 1/N\ \delta(x)$$

Extensión caso bidimensional

Para dos variables la transformada de Fourier discreta es:

$$F(u,v) = 1/MN \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp[-j2\pi(ux/M + vy/N)]$$

para u = 0, 1, ..., M–1 y v = 0, 1, ..., N–1

$$y\ f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp[j2\pi(ux/M + vy/N)]$$

para x = 0, 1, ..., M–1 e y = 0, 1, ..., N–1

Igual que en el caso unidimensional, f(x,y) representa muestras de la función $f(x_0 + x\delta x, y_0 + y\delta y)$ para x = 0, 1, ..., M–1 e y = 0, 1, ..., N–1. Análogamente también las relaciones entre los incrementos en los dominios del espacio y la frecuencia son:

$$\delta u = 1/M\delta x$$
$$\delta v = 1/N\delta y$$

Cuando se muestrea una señal, se obtiene una matriz cuadrada; por tanto, M = N, y entonces:

$$F(u,v) = 1/N \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp[-j2\pi(ux + vy)/N]$$

para u,v = 0, 1, ..., N–1, y:

$$f(x,y) = 1/N \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} f(u,v) \exp[j2\pi(ux + vy)/N]$$

para x,y = 0, 1, ..., N–1

Hay que observar en estas operaciones que se ha incluido el término 1/N, puesto que es una constante multiplicativa que no influye en el par F(u,v) y f(x,y) transformadas.



Propiedad de separabilidad

La transformada discreta de Fourier para el caso bidimensional, se puede expresar como:

$$F(u,v) = 1/N \sum_{x=0}^{N-1} \exp\,(-j2\pi ux/N) \sum_{y=0}^{N-1} f(x,y)\,\exp(-j2\pi vy/N)$$

para u,v = 0, 1, ..., N−1, y:

$$f(x,y) = 1/N \sum_{u=0}^{N-1} \exp\,(j2\pi ux/N) \sum_{v=0}^{N-1} F(u,v)\,\exp(j2\pi vy/N)$$

para x,y = 0, 1, ..., N−1

La ventaja de esta propiedad, es que F(u,v) o f(x,y) se pueden obtener en dos pasos por aplicación sucesiva de la transformada de Fourier unidimensional o su inversa. Esto se ve más claro si hacemos la siguiente separación de términos:

$$F(u,v) = 1/N \sum_{x=0}^{N-1} F(x,v)\,\exp[-j2\pi ux/N]$$

en donde,

$$F(x,v) = N[1/N \sum_{y=0}^{N-1} f(x,y)\,\exp(-j2\pi vy/N)]$$

En esta expresión, para cada valor de x, lo que nos encontramos entre corchetes es una transformada de Fourier unidimensional, con v = 0, 1, ..., N−1. Por tanto, la función bidimensional F(x,v) se obtiene haciendo una transformada unidimensional por filas sobre f(x,y) y multiplicando el resultado por N; por último, para obtener F(u,v), se realiza otra transformada unidimensional por columnas sobre F(x,v).

Nótese que se obtendrían los mismos resultados haciendo primero la transformada por columnas sobre f(x,y) y a continuación haciéndola por filas sobre el resultado.



Representación de la transformada de Fourier

Los espectros de muchas imágenes decrecen al aumentar la frecuencia, por lo que sus términos para frecuencias altas se hacen imperceptibles. Para compensar esto, y lograr una mejor visualización, se utiliza una representación logarítmica del tipo:

$$D(u,v) = \log[1 + K\,\mathrm{abs}(F(u,v))]$$

en lugar de F(u,v). La función D(u,v) no es nunca negativa.

Con esto se pretende que en el espectro de la función que representa una imagen todos los picos tengan una amplitud similar, de forma que se consiga el mismo brillo e intensidad en su representación. Vamos a demostrarlo viendo que, en efecto, la relación entre el máximo y el mínimo valor que puede tomar la función es menor en D(u,v) que en F(u,v). Sea:

$$R = \log(1 + F_{max})/\log(1 + F_{min})$$

donde F_{max} es el valor máximo alcanzado por F(u,v), y F_{min} es el mínimo de los valores que alcanza.

La relación F_{max}/F_{min} es siempre mayor que R. Por ejemplo, sea $F_{max} = 1.0$, y $F_{min} = 0.2$, tenemos entonces $F_{max}/F_{min} = 5.0$, y $R = 3.8$; si además la función F(u,v) la multiplicamos por una constante k para aumentarla, la relación F_{max}/F_{min} sigue valiendo 5.0, mientras que R, si $k = 20$, toma el valor 1.9, que es mucho menor.



Transformada rápida de Fourier

El número de multiplicaciones y sumas complejas que requiere el cálculo de la transformada discreta de Fourier de la expresión:

$$F(u) = 1/N \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N]$$

es proporcional a N^2, ya que son necesarias N multiplicaciones complejas de la expresión $f(x)\exp[-j2\pi ux/N]$ y (N–1) sumas. Por otro lado, los valores de la exponencial compleja pueden calcularse una sola vez y almacenarse en una tabla para posteriores cálculos.

Se ha propuesto un procedimiento de descomposición denominado transformada rápida de Fourier (FFT), que reduce el número de multiplicaciones y sumas a $N\log_2 N$, lo cual representa un coste computacional mucho menor que el que requiere la transformación directa, en especial cuando N es grande, como puede comprobarse en la siguiente tabla:

N	N^2	$N\log_2 N$
2	4	2
4	16	8
8	64	24
16	256	64
32	1024	160
64	4096	384
128	16384	896
256	65536	2048
512	262144	4608

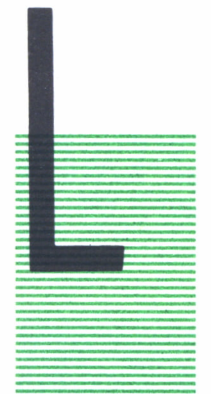
La mayor aplicación de las transformadas está en el diseño de filtros para mejora de imágenes y también en algunas técnicas de compresión de imágenes. En el apéndice 1 se describe con más detalle el algoritmo que permite el cálculo de la FFT. Asimismo, este algoritmo puede ser utilizado para evaluar la transformada de una imagen 2-D, efectuando, en primer lugar, una transformación por filas y posteriormente por columnas. En forma de pseudocódigo sería:

```
ALGORITMO FFT2D
FOR i:=1 TO N-1 DO
BEGIN
        X(i):=f(i,j)
        FFT(N,X)
        f(i,j):=X
    END;
    FOR j:=1 TO N-1 DO
    BEGIN
        X(j):=f(i,j)
        FFT(N,X)
        f(i,j):=X
    END.
```

Cada llamada para calcular la transformada rápida de Fourier comporta un coste computacional proporcional a $N\log_2 N$. Puesto que existen 2N de tales llamadas, el coste computacional total de la FFT bidimensional es proporcional a $N^2\log_2(N^2)$.

Este procedimiento supone un gran avance para la computación de la transformada de Fourier, sobre todo cuando N es un número muy grande. Por ejemplo, una implementación directa de la transformada de Fourier para N = 8192, necesita del orden de 45 minutos en una máquina como un IBM 7094; el mismo trabajo realizado con la implementación de la transformada rápida de Fourier emplearía cinco segundos en la misma máquina.

TECNICAS DE PROCESADO DE IMAGENES 3



LAS principales técnicas de procesado de imágenes pueden dividirse en dos grandes categorías: métodos en el dominio de la frecuencia y métodos en el dominio espacial. Los primeros se basan en efectuar la transformada de Fourier de una imagen, mientras que los segundos manipulan directamente los pixels de la imagen.

El fundamento de las técnicas en el dominio de la frecuencia es el teorema de convolución. Sea g(x,y) una imagen formada por la convolución de una imagen f(x,y) y un operador invariante espacialmente h(x,y), es decir:

$$g(x,y) = h(x,y) * f(x,y)$$

Entonces, por el teorema de convolución, tenemos la siguiente relación en el dominio de la frecuencia:

$$G(u,v) = H(u,v)\ F(u,v)$$

donde G, H y F son las transformadas de Fourier de g, h y f, respectivamente. La transformada H(u,v) es a veces llamada función de transferencia del proceso. Ambas ecuaciones son muy utilizadas en mejora y restauración de imágenes, siendo más importante la segunda, ya que la convolución discreta es a menudo realizada más eficientemente en el dominio de la frecuencia, mediante un algoritmo FFT.

En un problema típico de mejora, f(x,y) es dada y el objetivo, tras calcular F(u,v), es seleccionar un H(u,v) tal que la imagen deseada, dada por:

$$g(x,y) = F^{-1}\ [H(u,v)\ F(u,v)]$$

exhiba alguna característica importante de f(x,y). Por ejemplo, los bordes en la imagen f(x,y) pueden ser acentuados utilizando una función H(u,v) que enfatice los componentes de alta frecuencia de F(u,v).

Las técnicas en el dominio espacial se basan en transformaciones de los niveles de gris, donde el tipo de operación utilizada usada depende del criterio empleado para el realce o mejora, operando siempre sobre el valor de los pixels de la imagen.

Por otra parte, podemos hacer otra división de los métodos de mejora de imagen en el dominio espacial atendiendo al número de pixels de la imagen original, de los que depende el valor de cada pixel de la imagen resultante:

— Operaciones de punto o pixel individual: En ellas el valor de cada pixel de la imagen resultante es función unicamente del valor que tenía ese mismo pixel en la imagen original, mediante una transformación de algún tipo. Dentro de este grupo se encuentran las técnicas de modificación del contraste por ecualización del histograma de niveles de gris.

— Operaciones de área: En ellas se calcula el valor de cada pixel como una función de los valores de los pixels de un cierto entorno de dicho pixel en la imagen original y no sólo del valor original de ese pixel. En esta categoría se encuadran las operaciones de convolución y algunas transformaciones geométricas.



MANIPULACION DEL CONTRASTE

La manipulación del contraste es una transformación radiométrica puntual (pixel a pixel), cuyo objetivo es lograr una mejor discriminación de imágenes con bajo contraste. Cada nivel de gris de la imagen se modifica mediante la transformación seleccionada, sin considerar los valores de los niveles de gris de los pixels adyacentes. El análisis de los histogramas de una imagen constituye en la mayoría de las ocasiones un paso preliminar para lograr una eficiente manipulación del contraste de una imagen. El histograma de una imagen proporciona el porcentaje del número total de pixels que presentan un valor determinado de nivel de gris. El concepto es análogo al de densidad de probabilidad que se utiliza en estadística. Las características generales de los histogramas de niveles de gris en el caso de algunos tipos de imágenes aparecen en la figura 1.





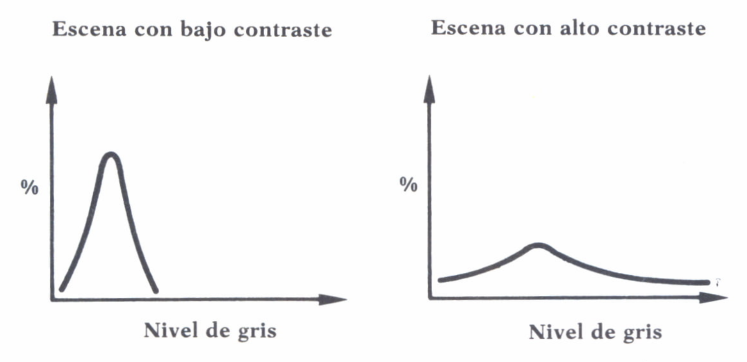


Fig. 1. Características de los histogramas de diferentes tipos de escenas.

El histograma de una imagen sólo especifica el número total de pixels de cada nivel de gris y no proporciona información acerca de la distribución espacial de los niveles de gris. La figura 2 presenta el resultado de la aplicación de una transformación de los niveles de gris en el histograma de una imagen.

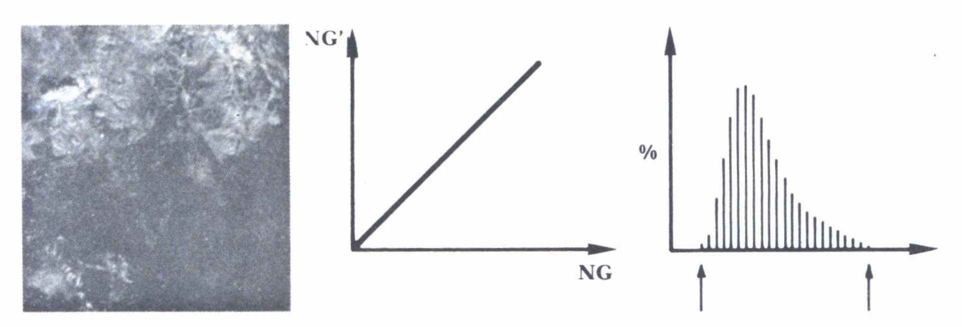


Fig. 2. Resultado de la aplicación de una transformación lineal en el histograma.

Cada nivel de gris de la imagen de entrada es proyectado en la curva de transformación, proporcionando un nuevo valor en la imagen de salida, conservándose el área del histograma obtenido. Esta simple operación lineal de mejora del contraste se denomina compresión del contraste. En ocasiones, y con objeto de conseguir un mejor contraste en la imagen de salida, se suele considerar algún tipo de saturación en ambos extremos del rango de salida, como se muestra en la figura 3.

Si el histograma es asimétrico, como sucede frecuentemente, es realmente complicado poder controlar mediante una simple transformación lineal tanto el valor medio del nivel de gris de la imagen de salida así como la cantidad de saturación en los extremos de la transformación efectuada.

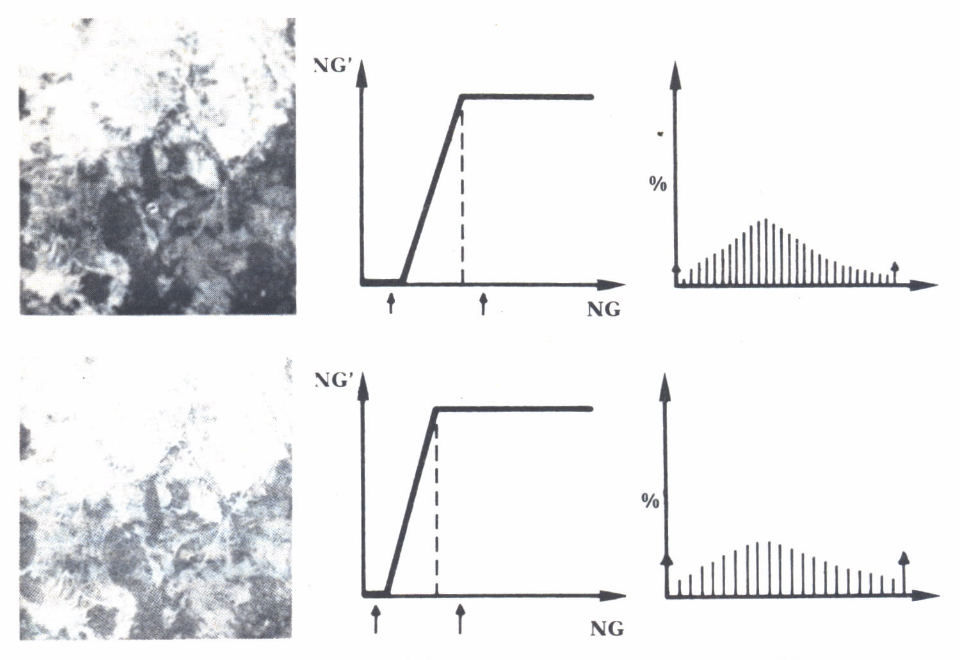


Fig. 3. Mejora lineal del contraste con saturación variable.

Es en estos casos en donde podría aplicarse una transformación lineal por tramos como la indicada en la figura 4.

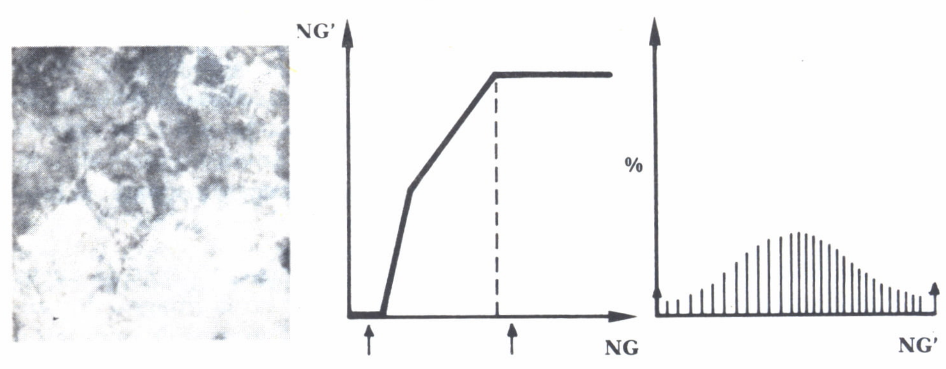


Fig. 4. Mejora del contraste no lineal.

Una operación de manipulación del contraste más general lo constituye la denominada ecualización del histograma, que se utiliza en una muy numerosa cantidad de imágenes. En este caso, se utiliza como función de transformación la distribución acumulativa de niveles de gris (el número total de pixels en el histograma entre cero y cada nivel de gris). La ecualización del histograma produce una reducción del contraste en áreas muy

claras o muy oscuras, expandiendo los valores intermedios. Un ejemplo de ecualización del histograma está representado en la figura 5.

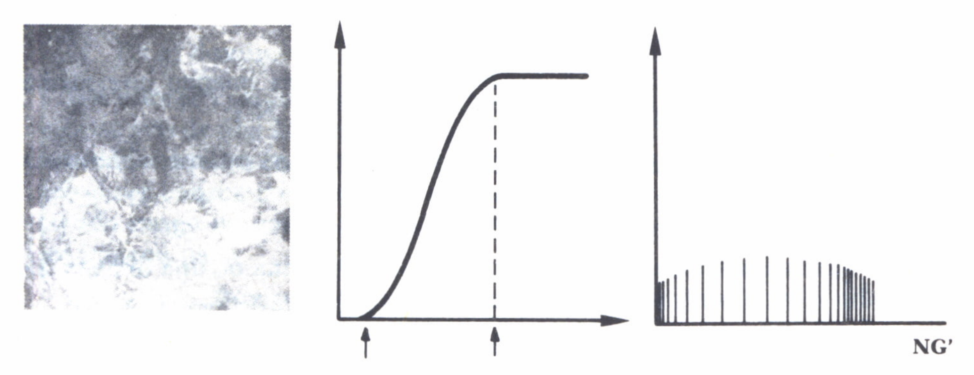


Fig. 5. Ecualización del histograma.

Entre los tipos de operaciones que pueden incluirse dentro de las de manipulación del contraste se encuentra la denominada como umbralización. El objetivo de la umbralización no es mejorar el contraste de la imagen considerada, sino segmentarla en dos clases diferenciadas: objetos y fondo, mediante la simple aplicación de un umbral en el rango de los niveles de gris. Para llevar a cabo la operación de umbralización supongamos que los niveles de gris de una imagen dada f(x,y) presentan el histograma de la figura 6.

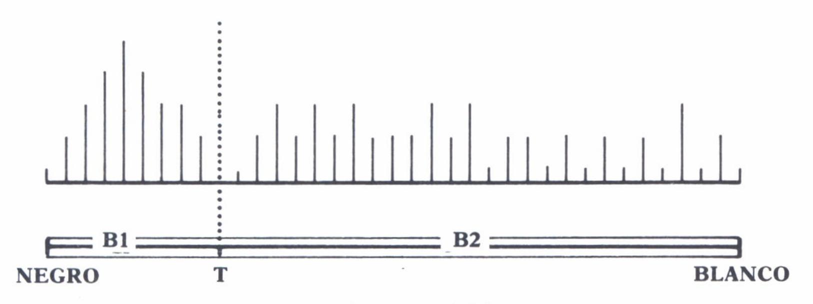


Fig. 6. Umbralización del histograma.

De la observación de la figura puede verse que la mayoría de los pixels de f(x,y) se encuentran concentrados en la zona más oscura, estando el resto repartidos más o menos uniformemente. Este histograma es característico de las imágenes que contienen objetos grises en un fondo oscuro. Para hallar la frontera entre los objetos y el fondo, dividiremos el histograma en dos zonas separadas por un umbral T. Se trata de seleccionar T de tal forma que la banda B1 contenga los niveles asociados al fondo, mientras que la banda B2 contenga los niveles de los objetos. Así, un cambio en el

nivel de gris de una banda a otra indicará la presencia de una frontera. Para conseguir la detección de fronteras tanto en sentido horizontal como vertical, será necesario explorar la imagen por filas y por columnas. Así pues, los pasos a seguir serían los siguientes:

a) Para cada fila de f(x,y), crear una fila de una imagen intermedia g1(x,y), utilizando la siguiente relación para y=0,1,...,N-1:

LE si los niveles de gris de f(x,y) y de f(x,y-1) g1(x,y) están en diferentes bandas
LB si se encuentran en la misma banda

b) Para cada columna de f(x,y), crear una columna de una imagen intermedia g2(x,y), utilizando la siguiente relación para x=0,...,N-1:

LE si g1(x,y) o g2(x,y) son iguales a LE g2(x,y)
LB si no son iguales a LE



SUAVIZADO DE IMAGEN

Las operaciones de suavizado se usan principalmente para reducir efectos espúreos que pueden presentarse en una imagen digital como consecuencia de un sistema de muestreo pobre o del canal de transmisión. Pueden realizarse tanto en el dominio espacial como en el de Fourier.

a) En el dominio espacial. Promedio de un entorno.

Dada una imagen f(x,y) se intenta generar una imagen suavizada g(x,y) cuyos niveles de gris en cada punto (x,y) se obtienen promediando los niveles de gris de los pixels de f contenidos en un entorno predefinido de (x,y). En otras palabras, la imagen suavizada se obtiene usando la relación:

$$g(x,y) = \frac{1}{M} \sum f(n,m) \qquad \forall ,(n,m) \in S$$

para x,y = 0,1,2,...,N-1. S es el conjunto de coordenadas de vecinos del punto (x,y), pero sin incluir a éste, y M es el número total de estos puntos.

El tamaño del entorno elegido es arbitrario, pero hay que tener en cuenta que el proceso de suavizado de bordes puede producir un efecto de «blurring» o emborronamiento en la imagen, que no es deseado. El grado de emborronamiento será, además, fuertemente proporcional al radio del entorno usado. Para un radio dado, el efecto de emborronamiento puede ser reducido usando un procedimiento de umbralización. Es decir, en lugar de usar la ecuación anterior para calcular g(x,y) en todos los puntos, usaremos dicha expresión si

$$f(x,y) - \frac{1}{M} \sum f(m,n) > T$$

y en caso contrario diremos que g(x,y) = f(x,y); donde T es un umbral no negativo predefinido. De esta forma podemos reducir el ruido de la imagen evitando en gran parte el efecto indeseado de «blurring».

b) En el dominio de Fourier. Filtros paso bajo.

Los bordes y otras transiciones pronunciadas (como el ruido) en los niveles de gris de una imagen, contribuyen grandemente al contenido de alta frecuencia de su transformada de Fourier. Por tanto, el emborronamiento podría ser conseguido, vía dominio de la frecuencia, atenuando un determinado rango de componentes de alta frecuencia en la transformada de una imagen dada.

Recordemos la ecuación:

$$G(u,v) = H(u,v)\ F(u,v)$$

donde F(u,v) es la transformada de la imagen que queremos suavizar. El problema es determinar una función H(u,v) que nos produzca G(u,v) por atenuación de las componentes de alta frecuencia de F(u,v). La transformada inversa de G(u,v) nos dará posteriormente la imagen suavizada deseada g(x,y). Ya que las componentes de alta frecuencia son «eliminadas» o filtradas y la información en el rango de bajas frecuencias es mantenida sin atenuación, este método se conoce comúnmente como filtrado de paso-bajo. La función H(u,v) se denomina función de transferencia del filtro considerado. En todos los casos que se considerarán más adelante, las operaciones de filtrado afectan de la misma manera a la parte real y a la parte imaginaria de la transformada de Fourier.



Ejemplos de filtrados paso-bajo

Los ejemplos siguientes corresponden a filtros paso-bajo con simetría rotacional, es decir, el filtro completo se generaría rotando 360° la sección mostrada en cada uno de los ejemplos señalados.

a) Rectangular. Un filtro paso-bajo cuya función de transferencia satisfaga la relación

$$H(R) = \begin{cases} 1 \text{ si } R \leq R0 \\ 0 \text{ si } R > R0 \end{cases}$$

Gráficamente, en la figura 7 se representa el perfil de este tipo de filtro, que suele denominarse «ideal», puesto que deja pasar todas aquellas frecuencias situadas dentro de un círculo de radio menor que R_0 sin atenuarlas, eliminando todas aquellas frecuencias situadas en el exterior del círculo. Al valor R_0 se le denomina frecuencia de corte, que constituye un

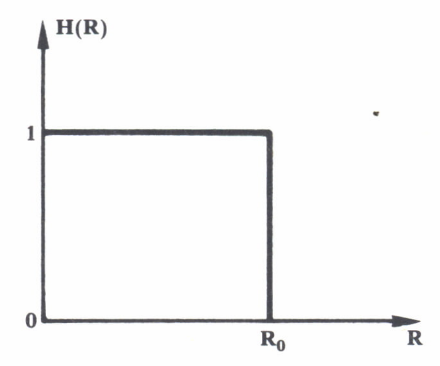


Fig. 7. Filtro paso-bajo ideal.

concepto muy útil a la hora de definir las características de un filtro, permitiendo la comparación con otros tipos de filtros.

El resultado de la aplicación de este tipo de filtros produce un emborronamiento de la imagen de partida como consecuencia de la eliminación de las altas frecuencias, o, lo que es lo mismo, una pérdida de definición de los bordes de la imagen.

b) Filtro de Butterworth

La función de transferencia de un filtro de Butterworth de orden n y de frecuencia de corte R_0, viene dada por la siguiente expresión:

$$H(R) = \frac{1}{1+[\ R/R_0\ ]2n}$$

Gráficamente, en la figura 8 se presenta el perfil de este filtro, pudiendo observarse cómo esta función de transferencia no presenta un cambio tan abrupto como en el anterior caso de filtro ideal, siendo en general más

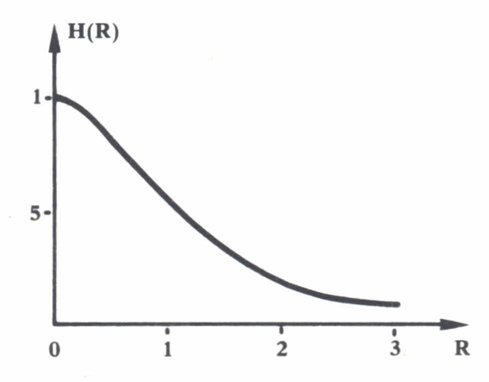


Fig. 8. Filtro paso-bajo de Butterworth.

aconsejable la utilización de este tipo de filtros que presentan una caída más suave, puesto que, como puede demostrarse, la utilización del caso ideal produce falsos contornos en la imagen filtrada. Seguidamente veremos algunos otros ejemplos de filtros que presentan una caída suave de la respuesta en frecuencias.

c) Filtro trapezoidal

Este tipo de filtro constituye un compromiso entre el filtro ideal y un filtro con caída suave. La función de transferencia de este filtro viene dada por la siguiente expresión:

$$H(R) = \begin{cases} 1 & \text{si } R < R0 \\ \dfrac{1}{R0-R1}(R-R1) & \text{si } R0 \leq R \leq R1 \\ 0 & \text{si } R \geq R1 \end{cases}$$

Gráficamente, el perfil de este tipo de filtro viene representado en la figura 9.

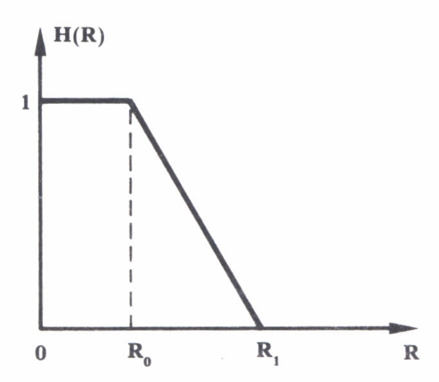


Fig. 9. Filtro paso-bajo trapezoidal.

d) Filtro exponencial

Este tipo de filtro paso-bajo, que también se utiliza frecuentemente en procesamiento de imágenes, presenta una frecuencia de corte situada a distancia R0 y una función de transferencia dada por la siguiente expresión:

$$H(R) = \exp[(R/R0)^n]$$

Gráficamente, el perfil de este tipo de filtros viene representado en la figura 10.

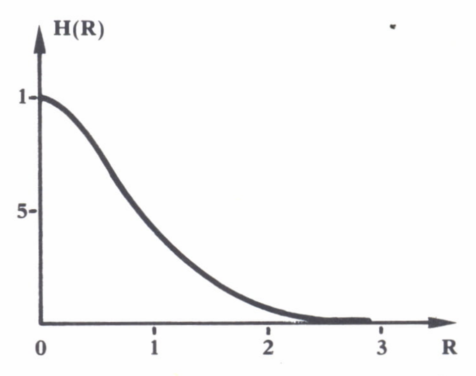


Fig. 10. Filtro paso-bajo exponencial.



REALCE DE BORDES

Puesto que, como ya se ha señalado, los cambios abruptos de los niveles de gris de una imagen están asociados con las componentes de alta frecuencia, la realización de operaciones de filtrado paso-alto producirá una atenuación de las componentes de baja frecuencia sin modificar sustancialmente la información de frecuencias altas contenida en el espectro de Fourier. Los ejemplos más usuales de filtros paso-alto, que presentan simetría rotacional, son análogos a los anteriormente estudiados:



Ejemplos de filtrados paso-alto

a) *Filtro rectangular*

Este tipo de filtros presenta una función de transferencia del tipo:

$$H(R) = \begin{cases} 0 \text{ si } R \leq R0 \\ 1 \text{ si } R > R0 \end{cases}$$

siendo R_0 la frecuencia de corte. Gráficamente, este tipo de filtros viene representado en la figura 11.

b) *Filtro de Butterworth*

La función de transferencia de un filtro paso-alto de Butterworth de orden n y frecuencia de corte R0 viene dada por la siguiente expresión:

$$H(R) = \frac{1}{1 + [R0/R]\ 2n}$$

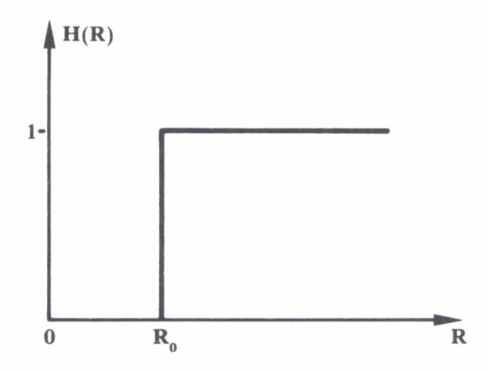


Fig. 11. Filtro paso-alto ideal.

Gráficamente, este tipo de filtros viene representado mediante la figura 12.

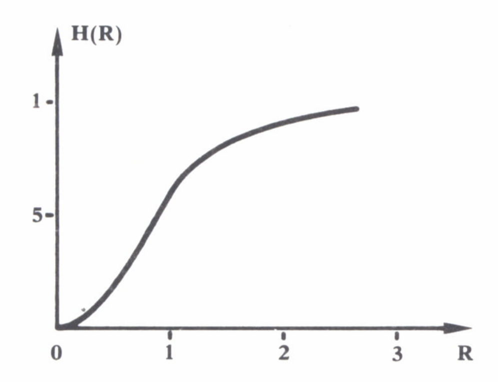


Fig. 12. Filtro paso-alto de Butteworth.

c) *Filtro exponencial*

La función de transferencia de un filtro paso-alto exponencial con frecuencia de corte R0, viene dada por la siguiente expresión:

$$H(R)= \exp[-R0/R]^n$$

en donde el parámetro n controla la velocidad de subida de la función exponencial. Una gráfica del perfil de este tipo de filtros viene dada en la figura 13.

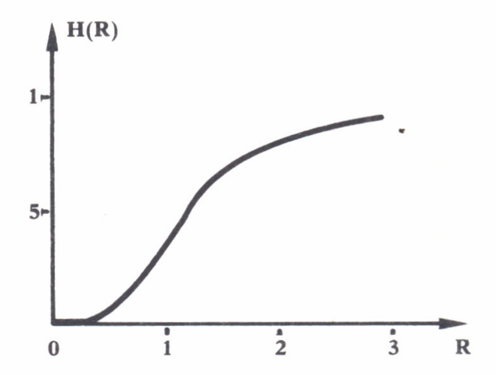


Fig. 13. Filtro paso-alto exponencial.

d) *Filtro trapezoidal*

La función de transferencia de un filtro paso-alto trapezoidal viene dada por la siguiente expresión:

$$H(R) = \begin{cases} 0 & \text{si } R < R1 \\ \dfrac{1}{[R0-R1]}[R-R1] & \text{si } R1 \leq R \leq R0 \\ 1 & \text{si } R > R0 \end{cases}$$

en donde R0 y R1 son parámetros conocidos y se cumple que R0>0R1. Gráficamente, este tipo de filtros viene representado en la figura 14.

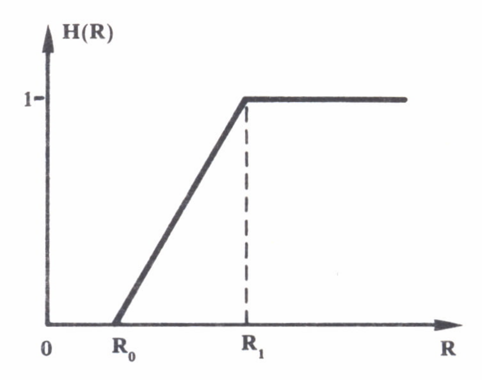


Fig. 14. Filtro paso-alto trapezoidal.



TRANSFORMACIONES GEOMETRICAS

En ocasiones es necesario efectuar cambios geométricos en una imagen digital, como, por ejemplo, cambiar de sistema de coordenadas, efectuar rotaciones, etc. Una vez realizada la transformación de coordenadas es necesario efectuar un proceso de interpolación con objeto de ajustar la imagen resultante a la retícula x,y, en donde se encuentra situada la imagen original. Por ejemplo, la transformación de coordenadas de un cuadrilátero definido por cuatro puntos puede efectuarse a través de los siguientes polinomios:

$$x = a_0 + a_1 x' + a_2 y' + a_3 x'y' + a_4 (x')^2 + a_5 (y')^2$$
$$y = b_0 + b_1 x' + b_2 y' + b_3 x'y' + b_4 (x')^2 + b_5 (y')^2$$

en donde las coordenadas con prima son las de la imagen de salida y las coordenadas sin prima corresponden a la imagen original. Si los coeficientes de $(x')^2$ e $(y')^2$ son cero, los ocho restantes coeficientes pueden calcularse a partir de los cuatro puntos correspondientes a los vértices del cuadrilátero de entrada. Los tipos de transformaciones posibles en este último caso vienen representadas en la figura 15.

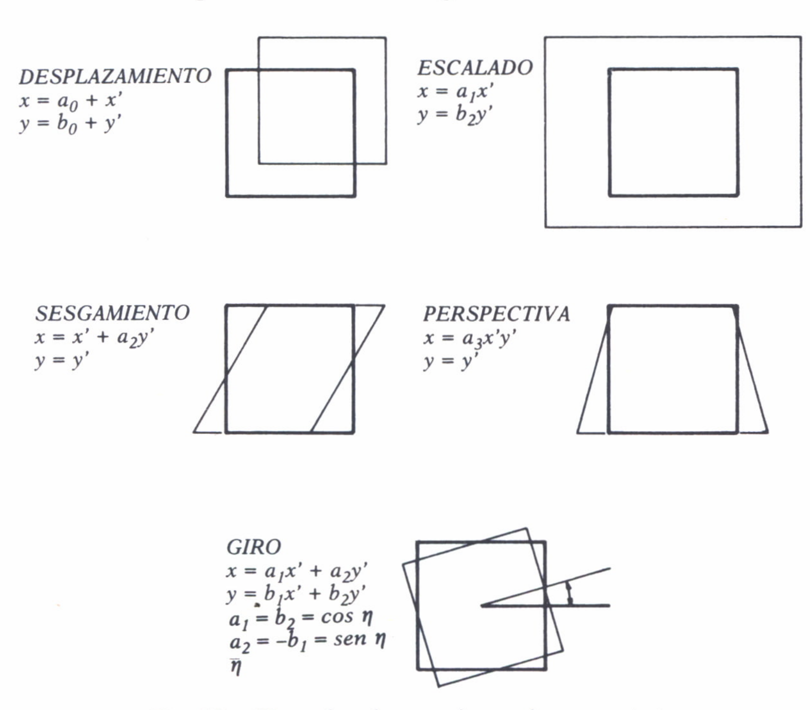


Fig. 15. Ejemplos de transformadas geométricas.

VISION ARTIFICIAL 4

LA mayor parte del desarrollo en visión artificial se ha llevado a cabo en aplicaciones industriales en donde el entorno en que se encuentra la máquina es bien conocido y la tarea encomendada al sistema de visión está claramente establecida. Un ejemplo típico lo constituye el sistema de visión utilizado por un robot para manipular objetos en movimiento situados en una cinta transportadora. En otras áreas de aplicación en donde la información a manipular está peor definida, el desarrollo llevado a cabo es comparativamente menor, como, por ejemplo, sucede en el caso del análisis e interpretación de imágenes de rayos X.

El objetivo de un sistema de visión artificial es analizar las imágenes y producir descripciones de dichas imágenes. El sistema de visión puede ser considerado como un elemento de un bucle de realimentación que está relacionado con la captación y análisis de imágenes, mientras que otros elementos del sistema están dedicados a la toma e implementación de dichas decisiones (figura 1).

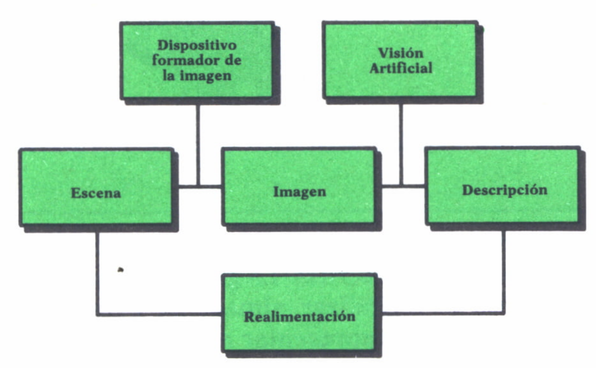


Fig. 1. Partes constituyentes de un sistema de visión artificial.

Existen diferentes niveles de descripción posibles. En general, se considerará suficiente aquel nivel de descripción que permita llevar a cabo la tarea deseada.

Entre las imágenes más sencillas que puede tratar un sistema de visión se encuentran las denominadas imágenes binarias. Una gran parte de las imágenes industriales que se procesan son binarias, puesto que en muchas ocasiones las descripciones que pueden extraerse de este tipo de imágenes son suficientes para este tipo de aplicaciones.

Una imagen binaria viene definida por una función característica b(i,j), que es cero para aquellos valores correspondientes al fondo de la imagen y uno para los puntos correspondientes al objeto. Un ejemplo de la imagen binaria está representada en la figura 2.

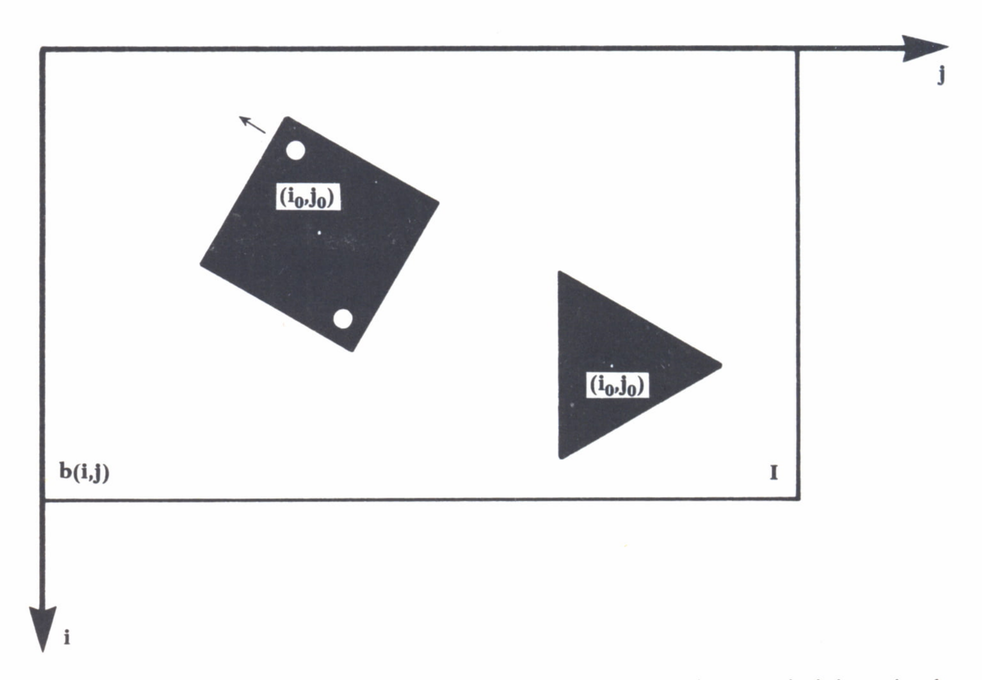


Fig. 2. Ejemplo de imagen binaria I, indicando los centros de gravedad de cada objeto.

Esta imagen binaria puede proceder de una operación de umbralización como las anteriormente descritas. La gran ventaja de las imágenes binarias es que son fáciles de digitalizar, almacenar y transmitir, puesto que con muy pocos bits es posible representar este tipo de imágenes.

El procesamiento de imágenes binarias requiere tan sólo el cálculo de

algunas propiedades geométricas y topológicas. Veamos seguidamente algunas de ellas:

$$\text{AREA:} \quad A = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b(i,j) \tag{4.1}$$

POSICION:

$$x_0 = 1/A \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} i\, b(i,j) \tag{4.2}$$

$$y_0 = 1/A \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} j\, b(i,j) \tag{4.3}$$

Las expresiones de la coordenada de la posición corresponde a lo que se denomina momentos de primer orden y coincide con el concepto de centro de gravedad de un objeto de la Mecánica.

Así como para el cálculo de la posición es necesario utilizar un momento de primer orden, para obtener la orientación es necesario el cálculo de momentos de segundo orden. Supongamos que se desea calcular la orientación del objeto que se encuentra situado en la figura 3. Para calcu-

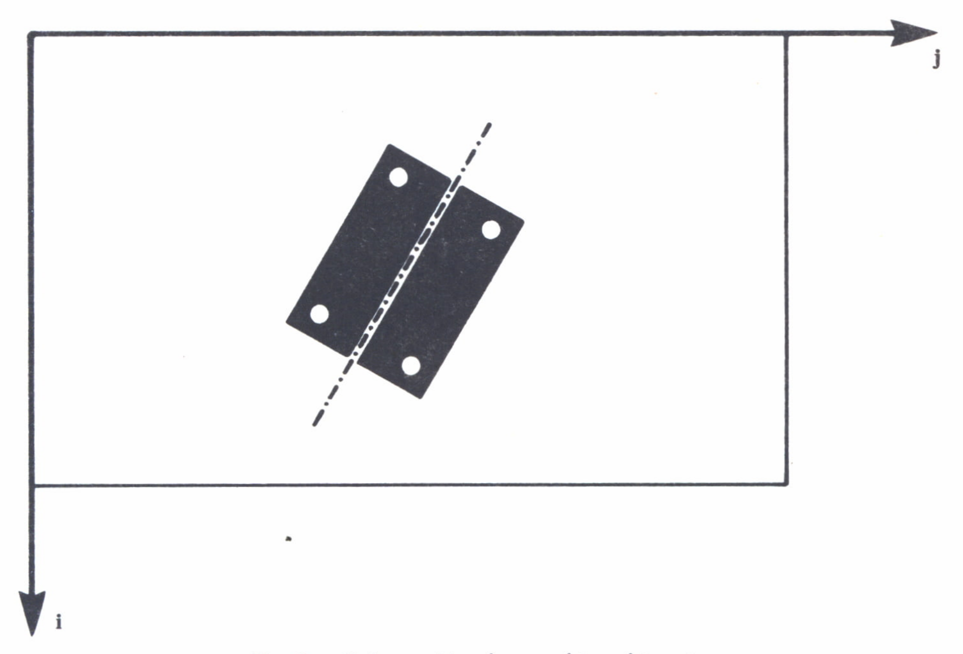


Fig. 3. Orientación de un objeto binario.

lar el eje según el cual se encuentra orientada la figura, hay que tener en cuenta que este eje coincide con el denominado eje de mínima inercia que se utiliza asimismo en Mecánica. Por ello, habrá que minimizar la siguiente expresión:

$$E = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} r^2 b(i,j) \qquad (4,4)$$

en donde r es la perpendicular trazada desde (i,j) a la línea anterior (figura 4). Puesto que el eje de mínima inercia pasa por el centro de gravedad, es

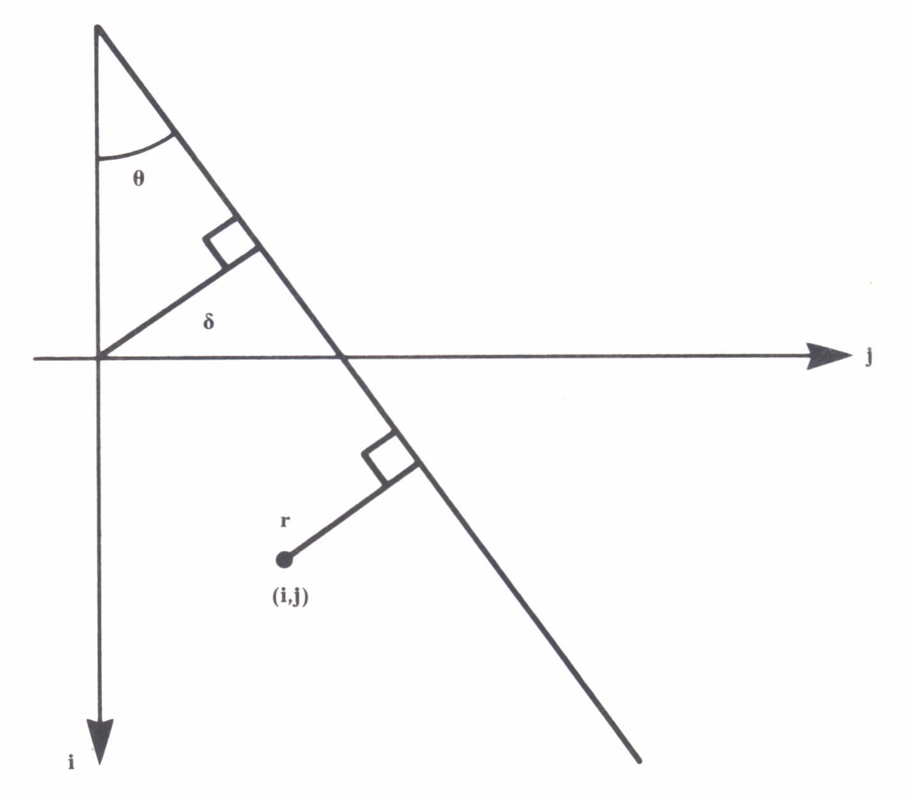


*Fig. 4. Distancia de un punto (i,j) a una lína: isen*θ *– jcos*θ + θ, *siendo* θ *la inclinación respecto al eje i y* θ *es la perpendicular a la línea desde el origen.*

conveniente efectuar un cambio del sistema de coordenadas a uno nuevo que pase por dicho punto y sea paralelo al antiguo:

$$i' = i - i_0;\ j' = j - j_0 \qquad (4.5)$$

Sustituyendo en la expresión (4.4) y teniendo en cuenta la distancia de un punto (i,j) a una recta definida por los parámetros δ y θ, de la Fig. 4 se obtiene la siguiente expresión:

$$E = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i' \text{ sen } \theta - j' \cos \theta)^2 \, b(i,j) \qquad (4.6)$$

y desarrollando la expresión anterior, queda:

$$E=(a \text{ sen}^2\theta\text{-}2b \text{ sen}\theta\cos\theta+c \cos^2\theta \qquad (4.7)$$

en donde:

$$a = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b(i,j) \, (i')^2 \qquad (4.8)$$

$$b = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b(i,j) \, (i'j') \qquad (4.9)$$

$$c = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b(i,j) \, (j')^2 \qquad (4.10)$$

Para hallar la orientación θ_0 del eje de mínima inercia habrá que derivar e igualar a cero la expresión (4.7), obteniendo como resultado:

$$\theta_0= 1/2 \text{ tang}^{-1}(2b/a\text{-}c) \qquad (4.11)$$

En la figura 5 se presenta un procedimiento en lenguaje LISP denominado CENTRO, que determina el centro de gravedad de una imagen rectangular de dimensión N filas x M columnas, devolviendo los valores I_o y J_0 de las coordenadas del centro de gravedad. En la figura 6 se presenta un procedimiento LISP que devuelve una lista que contiene el centro de gravedad, así como el valor del ángulo que forma el eje de mínima inercia con el eje (i).

```
(DEFUN CENTER (IMAGE)
  (LET ((N (ARRAY-DIMENSION IMAGE 0))             ;N filas.
        (M (ARRAY-DIMENSION IMAGE 1))             ;M columnas.
        (SUM 0) (SUM-I 0) (SUM-J 0)              ;Acumuladores.
    (DO  ((I 0 (+ I 1)))                          ;Un paso a través
                                                   de las filas.
         ((= I N)                                 ;¿Recorridas to-
                                                   das las filas?.
```

```
        (COND((ZEROP SUM) 'NO-OBJECT)              ;Todo cero, sino
            (T (LIST(/(FLOAT SUM-I) (FLOAT SUM)    ;Procesar i-O,
                 (/(FLOAT SUM-J) (FLOAT SUM))))) ;Procesar j-O.

    (DO ((J O (+ J 1)))                            ;Un paso a través
                                                    de las columnas.

        ((= J M ))                                 ;¿Recorridas todas
                                                    las columnas?.

      (COND((ZERO(AREF IMAGE I J)))                ;Ignorar ceros.

          (T (SETQ SUM (+ SUM 1)                   ;Total de p(i,j), &

                   SUM-I (+ SUM-I I)               ;total i*p(i,j),

                   SUM-J (+ SUM-J J))))))))        ;total j*p(i,j).
```

Fig. 5. Función LISP que permite calcular el área de una imagen binaria de dimensión N × M.

```
(DEFUN ORIENTATION (IMAGE)

  (LET ((N (ARRAY-DIMENSION IMAGE O))              ;N filas.

        (M (ARRAY-DIMENSION IMAGE 1))              ;M columnas.

        (C-O-A (CENTER IMAGE)))                    ;Centro de área.

    (COND   ((ATOM   C-O-A)  C-O-A)                ;Ningun objeto
                                                    encontrado.

          (T (LET ((IO (CAR C-O-A)) (JO (CADR C-O-A))

                   (A O O) (B O O) (C O O)         ;Acumuladores'.

               (DO ((I O (+ I 1)))

                   ((= I N)                        ;Un paso por fila.

                   (COND ((AND (ZERO B) (= A C)) ;Simetria.

                          (APPEND C_O_A ,(SYMETRIA)))

                         (T (APPEND C-O-A          ;Dirección de ejes.

                               (LIST (/ (ATAN (* 2.0 B)

                                         (- A C)) 2.0 ))))))

               (DO ((J O (+ J 1 ))) ((= J M))  ;Un paso por columna

                 (COND ((ZERO (AREF IMAGE I J))) ;Ignora ceros.

                   (T (SETQ A (+ A (* (- I IO) (- I IO)))

                          B (+ B (* (- I IO) (- J JO)))

                          C (+ C (* (- J JO) (- J JO)))))))))))))
```

Fig. 6. Función LISP que permite el cálculo de la orientación de una imagen binaria N × M.

Otras características geométricas de interés pueden ser el perímetro de la figura estudiada, que puede estimarse como el número de ceros que se encuentran contiguos con unos. En la figura 7 se presenta un procedimiento que permite calcular el perímetro de una imagen binaria, si bien cabe señalar que el valor obtenido es una sobreestimación del valor real, debido a que el proceso de cuantificación produce en la frontera de los objetos un efecto en diente de sierra.

```
(DEFUN PERIMETER (IMAGE)
  (LET ((N ARRAY-DIMENSION IMAGE 0))              ;N filas.
        (M ARRAY-DIMENSION IMAGE 1))              ;M columnas.
        (SUM 0))                                  ;Acumulado total.
   (DO (( I 1 (+ I 1))) ((= I N) SUM)             ;Un paso
                                                   por fila.
    (DO ((J 1 (+ J 1))) ((= J M))                 ;Un paso por
                                                   columna.
        (COND ((NOT (= (AREF IMAGE I J)
                       (AREF IMAGE I (- J 1))))
               (SETQ SUM (+ SUM 1))))                ;0-1 & 1-0
                                                      vertical.
        (COND ((NOT (= (AREF IMAGE I J)
                       (AREF IMAGE (- I 1) J)))
               (SETQ SUM (+ SUM 1))))))))            ;0-1 & 1-0
                                                      horizontal.
```

Fig. 7. Función LISP que permite calcular el perímetro de una imagen binaria de dimensión N × M.

SISTEMAS DE PROCESAMIENTO DE IMAGENES 5

CARACTERISTICAS GENERALES

LOS sistemas de procesamiento de imágenes se definen como aquellos capaces de realizar alguna de las siguientes operaciones:

a) Adquirir imágenes de uno o más sensores, representándose éstas en forma de imagen. La representación de una imagen se efectúa mediante una matriz bidimensional, constituyendo el nivel de gris de cada pixel la tercera dimensión.

b) Almacenar imágenes de manera que puedan ser procesadas.

c) Realizar funciones de procesado tales como mejora, restauración, codificación, análisis, etc.

Los componentes necesarios que requieren los sistemas de procesamiento para llevar a cabo sus funciones son los siguientes:

— SENSORES, como, por ejemplo, cámaras de TV vidicón, cámaras de TV de estado sólido CCD, etc.

— INTERFACES DE CAPTACION DE IMAGENES (FRAME GRABBERS).

— MEMORIAS DE IMAGEN, para almacenamiento y presentación.

— PROCESADORES DE IMAGEN, incluyendo los denominados procesadores puntuales y los procesadores espaciales (array processors).

— MONITORES.

— MEMORIAS DE ALMACENAMIENTO.

— INTERFACES DEL USUARIO CON EL SISTEMA.

— SOFTWARE BASICO, que permita al usuario la realización de operaciones de procesado.



Descripción de los componentes del sistema

a) *Sensores*

La tecnología estándar para la imagen electrónica es la cámara vidicón. La mayoría de las cámaras vidicón incluyen la electrónica necesaria para producir una imagen de TV convencional. La sensibilidad espectral del vídeo va desde 350 a 700 nanómetros. La luz es enfocada en un objetivo fotosensible que genera un modelo de imagen como carga, objetivo que es explorado por un haz electrónico produciendo la señal de vídeo. El haz electrónico se genera en un tubo de vacío, lo cual hace que este tipo de dispositivos sean pesados y frágiles, requiriendo tensiones elevadas (900 V).

En las cámaras que utilizan tecnología de estado sólido (CCD, siglas de Charged Coupled Devices), la luz que incide sobre cada elemento genera una carga proporcional a la intensidad lumínica, recogiendo esta carga mediante condensadores, y transferida en forma de paquetes de carga a un amplificador. La imagen aparece entonces como una serie de voltajes a la salida del amplificador. Las ventajas más importantes de las cámaras CCD son su gran robusted mecánica, su alta sensibilidad, difícil saturación, así como un peso y un tamaño reducidos. Sus desventajas más importantes son su coste elevado, así como su baja resolución. En la actualidad y debido a sus características, se tiende a utilizar en entornos industriales cada vez en mayor medida las cámaras CCD, puesto que en este tipo de aplicaciones la baja resolución no impide llevar a cabo las tareas deseadas.

b) *Interfaces de captación de imágenes (frame grabbers)*

En la mayoría de las ocasiones la señal de vídeo procedente de las cámaras de TV es digitalizada y almacenada en el computador como una matriz bidimensional de niveles de gris. La función de un «frame grabber» es la de convertir la señal de entrada en un formato que pueda ser almacenado y procesado. Este proceso requiere la digitalización de la señal y su transferencia a la memoria de imagen. Los componentes de un «frame grabber» son los siguientes:

— Adquisición de la señal de entrada.
— Sincronización con el formato de la señal de entrada.
— Convertidor analógico-digital de alta velocidad.

Es importante tener en cuenta el formato de las señales de entrada a la hora de efectuar su captación, ya que algunas señales, como las de TV, incorporan los sincronismos en la señal de vídeo, mientras que en otro tipo de señales los sincronismos aparecen exteriormente.

c) *Memorias de imagen*

Las imágenes procedentes del sensor se almacenan en la memoria de imagen en posiciones consecutivas, representando cada valor almacenado el nivel de gris de la imagen considerada. Debido al gran volumen de información necesario para almacenar imágenes, las memorias de imagen son de gran capacidad. Así, por ejemplo, una imagen de 512 × 512 pixels × 8 bits de cuantificación requiere 0.25 Mbytes de almacenamiento. En cuanto a los tiempos de acceso, es decir, el tiempo requerido para leer o escribir en la memoria de imagen, no debe exceder el requerido para la captación de la señal de vídeo. Este hecho supone que en algunos sistemas de procesado de imágenes la propia memoria de refresco se utilice como memoria de imagen.

La memoria de refresco constituye la parte más importante del sistema de visualización, ya que su diseño es crítico. En los visualizadores recientes, el hecho de que la memoria de refresco se utilice como memoria de imagen, permite realizar funciones tales como ampliación, desplazamiento, corte, etc. Estas operaciones no destruyen los datos contenidos en la memoria. En la mayoría de los sistemas, la memoria de refresco se encuentra diseñada en forma de canales. Un canal típico contiene una imagen de 512 × 512 × 8 bits, lo cual equivale a refrescar una imagen monocroma en pantalla. Si se utilizan tres canales en paralelo, podrían representarse imágenes en color. Los sistemas actuales proporcionan entre 1 y 64 canales, existiendo asimismo sistemas que proporcionan imágenes de 1024 × 1024 pixels y 16 bits de precisión.

El computador central accede a la memoria de refresco mediante alguno de los siguientes métodos: 1) a través de una interface que permite acceder al usuario de forma aleatoria por medio de una transferencia DMA, o 2) considerando una subregión rectangular del canal dentro del espacio de direccionamiento del computador, por lo que se accede a la misma como si se tratase de la memoria principal del computador.

Desde mediados de los setenta, las memorias de refresco se han construido utilizando memorias RAM. Las sucesivas mejoras en la tecnología RAM ha permitido incrementar la capacidad de los chips, por ejemplo, 4 Kbits, 16 Kbits, 64 Kbits, etc. Sin embargo, existe una relación inversa entre el tiempo de acceso a este tipo de memorias y el consumo de potencia, porcentaje de errores y coste. Es por ello que las memorias de refresco se han construido a partir de dispositivos relativamente lentos. Un valor típico del tiempo de ciclo de este tipo de memorias es el de 400 nseg., que es mucho más lento que el requerido para el refresco de los visualizadores. Para obtener las velocidades de refresco necesarias, las memorias de los sistemas utilizan multiplexación de pixels, típicamente con un factor de 8 en los visualizadores que utilizan un formato de 512 × 512. En general, la mayoría de los sistemas multiplexan cada bit de intensidad en un chip di-

ferente; por ello, en un diseño típico se accede a 64 chips de RAM durante un simple ciclo de memoria.

La mayoría de las limitaciones de las memorias de refresco proceden de la utilización de la multiplexación. Este hecho supone que algunos visualizadores no dispongan de todos los valores enteros de ampliación (zoom), siendo los valores 2:1, 4:1 y 8:1 mucho más fáciles de implementar. La multiplexación también limita el acceso por columnas, puesto que los ocho pixels multiplexados son direccionados por filas. La consecuencia de ello es que el acceso por columnas será ocho veces más lento que por filas.

d) *Monitores de presentación de imágenes*

El monitor de presentación de imágenes constituye un monitor industrial de alta calidad. Generalmente, se encuentra calibrado para presentar una trama cuadrada de datos con una relación de aspecto 1:1. Los tamaños de presentación estándar son 512 × 512 o 1024 × 1024 pixels, si bien también suelen encontrarse otros tamaños tales como 640 × 480, 1024 × 1280. La señal de TV comercial utiliza velocidades de 30 (o 25) tramas/seg., con una relación de aspecto 2:1 y entrelazamiento. Sin embargo, se vienen utilizando recientemente visualizadores con un refresco de 60 (o 50) tramas/seg. y sin entrelazamiento. Este último tipo de visualizadores produce una presentación carente de parpadeos, lo cual minimiza la fatiga del observador.

Por otro lado, es bastante frecuente la utilización por parte de algunos fabricantes de terminología ambigua, fundamentalmente en el caso del término resolución. La única medida de resolución aceptable consiste en colocar un test de prueba en el monitor y medir ópticamente la resolución con un fotómetro.

Las principales restricciones de los visualizadores se deben a la divergencia del haz de electrones y a la dispersión de los fotones. En la figura 1 se presenta la arquitectura típica de un visualizador de un sistema de procesamiento de imágenes.

e) *Procesadores de imagen*

Existen un gran número de arquitecturas para procesamiento de imágenes, incluyendo las arquitecturas especializadas. En general, cabe señalar que los procesadores serie convencionales tipo von Neuman no son los más adecuados para el procesado de imágenes, sino más bien arquitecturas de tipo paralelo.

Las principales arquitecturas para procesamiento de imágenes son las siguientes:

— Procesadores serie convencionales.
— Procesadores con paralelismo lineal (pipeline).

Computador Central

Interface del computador

Procesador de control y/o controlador de hardware

Interface hombre-máquina

Refresco de memoria virtual

Memoria de refresco

Punto de procesador

Presentación de la visualización

Monitor de imagen

Procesador espacial

Interface de vídeo externo

Vídeo analógico

Fig. 1. Arquitectura de un visualizador de un sistema de procesamiento de imágenes.

— Procesadores paralelo (array processors).
— Procesadores de propósito especial.
— Procesadores VLSI.

Como anteriormente se indicó, los procesadores serie no son los más adecuados para el procesado de imágenes. Sin embargo, presentan ventajas sustanciales con el resto de procesadores, como son su bajo coste y su flexibilidad.



Métodos de iluminación

Un aspecto importante a tener en cuenta a la hora de efectuar la captación de una imagen por parte de un sistema de visión lo constituye el método empleado para la captación de dicha imagen. Cuando lo que se desea es captar una escena industrial es necesario, en primer lugar, decidir el número y la configuración de las cámaras a emplear, puesto que si se desea

obtener información tridimensional, será necesario el concurso de varias cámaras. Otro de los aspectos a tener en cuenta es el método de iluminación empleado, puesto que una iluminación inadecuada puede producir sombras, brillos u otros efectos indeseados. Los métodos más usuales de iluminación de escenas son los siguientes:

a) *Iluminación difusa*. Es un tipo de iluminación indirecta que conduce a una buena eliminación de sombras, permitiendo una modelización tridimensional del objeto considerado utilizando una sola cámara (figura 2).

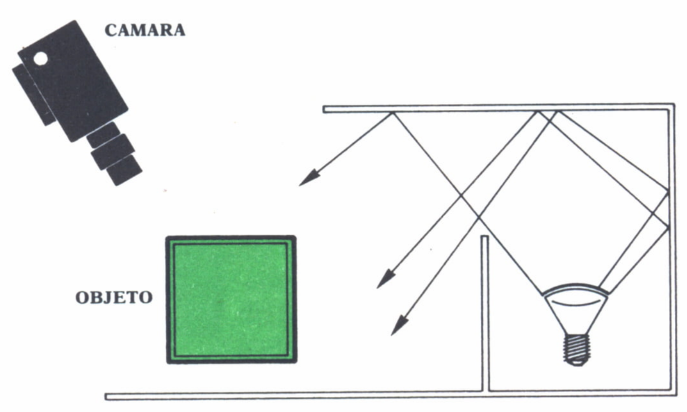


Fig. 2. Ejemplo de disposición de una iluminación difusa.

b) *Iluminación posterior*. Constituye el sistema de iluminación más utilizado. Permite el reconocimiento de piezas pequeñas y ligeras, por lo que es un método muy indicado para aplicaciones de control de calidad y clasificación de piezas (figura 3).

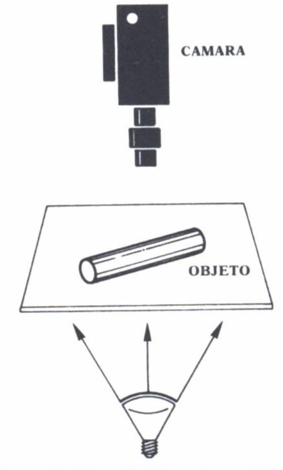


Fig. 3. Disposición de iluminación posterior.

c) *Iluminación modulada*. Este tipo de iluminación permite la extracción de información de curvaturas, así como de su uniformidad superficial por medio de la deformación que experimentan unas líneas proyectadas sobre el objeto, mediante la utilización de una máscara conocida (figura 4).

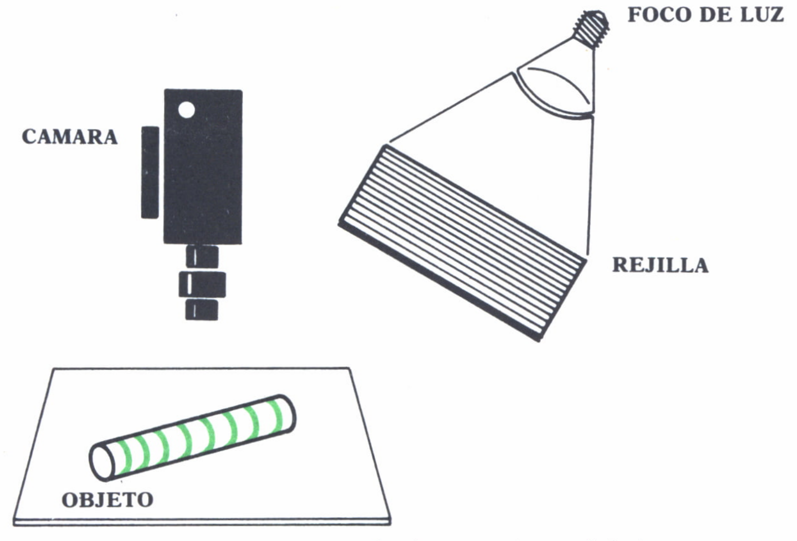


Fig. 4. Disposición de iluminación modulada.

d) *Iluminación direccional*. Este tipo de iluminación se utiliza fundamentalmente para la observación de las rugosidades superficiales de un objeto, así como para la observación de pequeñas áreas del objeto considerado, mediante la utilización de un haz coherente (láser) o no (figura 5).

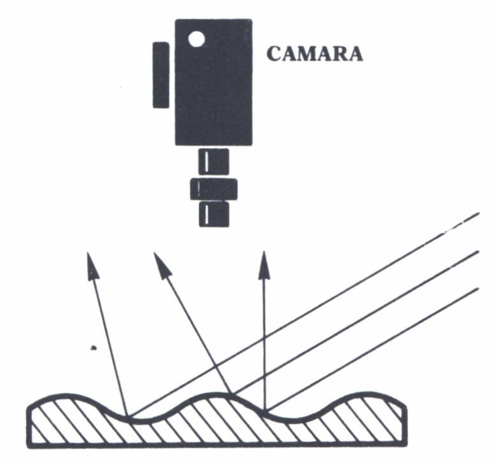


Fig. 5. Disposición de iluminación direccional.

CARACTERISTICAS HARDWARE Y SOFTWARE DE ESTOS SISTEMAS

En los últimos diez años, los sistemas para procesamiento de imágenes han experimentado un considerable desarrollo, de tal manera que operaciones de procesado que en los primeros sistemas suponían un elevado tiempo de cálculo, en la actualidad se realizan prácticamente en tiempo real. A partir de los años setenta, estos sistemas se han venido utilizando fundamentalmente en investigación. La razón fundamental del interés por el diseño de este tipo de sistemas radica en el hecho de que la tecnología serie de los computadores existentes no es la más adecuada para el procesamiento de este tipo de información.

Los primeros intentos de construcción de este tipo de sistemas se efectuaron en la década de los cincuenta. Estos sistemas se caracterizaban por su gran tamaño, memoria limitada y mala calidad de imagen, debida al ruido. El procesamiento digital de imágenes comenzó realmente a fines de los sesenta, con el desarrollo de computadores y minicomputadores más baratos que los anteriores. Un ejemplo del software de procesamiento de imágenes desarrollado en esa época es el VICAR, construido por el Jet Propulsion Laboratory. Este tipo de sistemas estaban basados en un ordenador central y las tareas se atendían mediante una cola de trabajo y eran de un coste excesivamente caro como para poder ser comercializados. Algo más tarde, el Laboratorio de Procesamiento de Imágenes, de la Universidad del Sur de California, efectuaba los primeros procesamientos locales con ayuda de un microcomputador.

La arquitectura propuesta a principios de los setenta, basada en un minicomputador junto a una serie de periféricos, se ha mantenido hasta la actualidad. Los periféricos utilizados permiten la adquisición y digitalización de las imágenes, su presentación y registro. Por otro lado, y con objeto de aumentar la velocidad de procesamiento, se les suele conectar a un computador central o incluso añadirles un procesador de array.

Los primeros visualizadores comerciales fueron el Aerojet (posteriormente denominado Comtal) y permitían la realización de manipulaciones radiométricas, superposición de gráficos y acceso directo a los pixels de la imagen mediante cursor. Estos visualizadores estaban limitados a la presentación de las imágenes y a manipulaciones globales de la intensidad o del color.

En la mitad de los setenta, se desarrollaron los primeros sistemas de procesamiento realmente interactivos con la incorporación de software de tiempo real. Así, por ejemplo, cabe citar el sistema VICAR, que posteriormente dio lugar al Mini-VICAR, el sistema GE 100, de General Electric para procesamiento interactivo de imágenes LANDSAT, o el sistema PECOS, de Electromagnetic Systems Laboratory. La evolución de las nuevas

tecnologías de computadores fue acompañada de una progresiva evolución de los sistemas de presentación. Así, la incorporación de memorias de acceso aleatorio RAM como memorias de refresco permitió la realización de operaciones más complejas, como, por ejemplo, la suma y resta efectuadas a velocidad de vídeo; el cálculo de histogramas por hardware, particiones y ampliaciones de la imagen, etc.

A mediados de los setenta, International Imaging Systems introdujo el denominado Modelo 70, que incorporaba un bucle de realimentación a velocidad de vídeo, que con el paso de los años se convirtieron en una característica común a todos los equipos, incorporando unidades aritmético-lógicas para aumentar la velocidad (como, por ejemplo, los sistemas Gould DeAnza, Vicom, Ikonas, etc). El procesamiento en forma de bucle de realimentación refleja una de las características básicas de los algoritmos de procesamiento de imágenes: la realización de operaciones simples con imágenes puede ser aplicada iterativamente a cada uno de los pixels.

Los visualizadores desarrollados a fines de los setenta y principios de los ochenta incorporaban las siguientes características: 1) operaciones opcionales para la realización de operaciones específicas, tales como operaciones de convolución, y 2) arquitecturas paralelas y bucles de realimentación para la realización de algoritmos iterativos.



Sistemas comerciales de procesamiento digital de imágenes

Los sistemas comerciales de procesamiento digital de imágenes pueden ser de tres tipos:

• *Sistemas de propósito general.* Están constituidos por un minicomputador, junto con los periféricos necesarios para efectuar las operaciones de procesamiento. En otras ocasiones se comercializa sólo el procesador de imágenes para ser conectado a algún computador central. Ejemplos de este tipo de sistemas lo constituyen el sistema VICOM y el GOULD DeAnza. En la figura 6 aparece un esquema de la arquitectura del sistema de procesamiento de imágenes VICOM. Como caso particular de este tipo de sistemas se encuentran los denominados de propósito específico. Constituyen sistemas cerrados utilizados generalmente para aplicaciones concretas. Así, por ejemplo, cabe citar el sistema InterScan, de Control Automation. En la figura 7 aparece un cuadro comparativo de algunos de los sistemas comerciales existentes, tanto de propósito general como de propósito específico.

• *Estaciones de ingeniería (workstations).* Este tipo de sistemas proporcionan una potencia local de cálculo elevada, de tal manera que el usua-

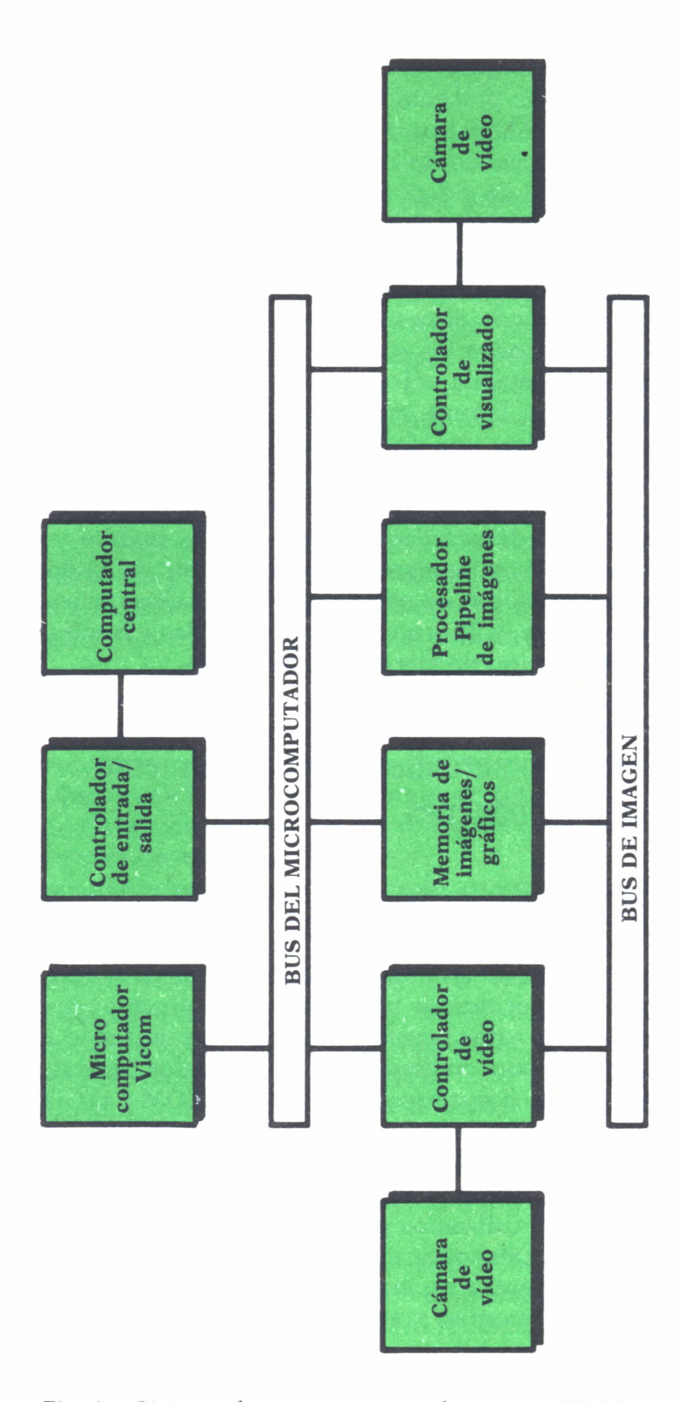


Fig. 6. Sistema de procesamiento de imagen VICOM.

		CAMARA			MICROPROCESADOR	
MARCA	MODELO	VIDICON	ESTADO SOLIDO	NUMERO CAMARAS	TIPO	BITS
Vicom	Vicom	X		1	68000	16
Kontron	IPS I	X		1	Z80	8
	IPS II	X		1	Z80	8
Digital Design	Visionix		Fotodiodos	1	68000	16
Control Automation	Intervision TM 2000		X	64	8086	16
	Intervision TM 1000		X	4	8086	16
Automation	CAV-1000 Vision System		CCD	4	8086	16
GS Beam Positioning Comp.	gs/Eye Configuration Two	X		1		16
Icos N.V.	Icos 2000 System	X	X	2	68000	16
Foxville	Digi/Cam Imagin System		X	1	—	—
Ors	Ors Scan System Model 100	X		1	—	8
Micron Technology, Inc	Micron Eye		X	1	6502	8
Fairchild	Sistema con cámara CCD 3000		CCD	1	F9445	16
TS Products	Inspector	X		1	—	—
i-hot Vision Systems, Inc	i-hot I		CCD	1	8086	16
Liacom	CVD	X		1	8 x 305	8
	GMD 02	X		1	8 x 305 + Z80	8
	HRD	X		1	8 x 305	8
Aydin	AY 52000	X		1	8086	16
Cubicomp	CS-5	X		1	—	—
Eikonix			X	1	—	8

Fig. 7a. Cuadro comparativo de sistemas comerciales de procesamiento digital de imágenes.

PROCESADO	VISUALIZACION, RESOLUCION				CARACTERISTICAS
VELOCIDAD	B/N	COLOR	GRISES	PIXELS	
–	X	X	256	256 x 256	Tiempo real, convolución espacial Programas en ASSEMBLER, FORTRAN, PASCAL Aplic. en convolución, inspec. y análisis
13 seg.	X		256	512 x 256	Programas en BASIC y FORTRAN Inspección y microscopía
6,5 seg.	X		256	512 x 256	Programas en BASIC y FORTRAN Inspección y microscopía
10 seg.	X		256	256 x 256	Tiempo real Programas en FORTRAN y PASCAL Inspección de objetos móviles, control de medid.
–	X		64	320 x 484	Programa en FORTH Aplicaciones de inspección
–	X		64	128 x 128	Programa en FORTH Inspección
–	X		80	128 x 128	Tiempo real Robótica, inspección
–	X		20	–	Posicionamiento de componentes en inspección Robótica
–	X		256	2000 x 2000	Inspección visual
–	X		4	128 x 256	
45 seg.	X		–	–	Inspección visual
–	X		–	256 x 256	
–	X		–	256 x 256	Accionamiento de formas, inspección Robótica
10 seg.	X	X	–	–	Programa en BASIC Aplicaciones de inspección y microscopía
3 seg.	X		256	256 x 240	Robótica
–	X	X	256	256 x 256	Tiempo real Aplicaciones en medicina, mezcla de imágenes inspección
–	X	X	256	512 x 512	Tiempo real Aplicaciones en medicina, mezcla de imágenes, inspección
–	X	X	256	1024 x 1024	Tiempo real Amplía el GVDO2 y el GMO2 Aplic. en medicina, mezcla de imágenes, inspección
7 seg.	X	X		2048 x 2048	Tiempo real Programas en FORTRAN y BASIC Aplicaciones en medic., topograf., análisis
–		X	–	512 x 512	Inspección
17 seg.	X		256	1024 x 1024	Matriz lineal Inspección y microscopía, medicina

Fig. 7b. Continuación del cuadro comparativo de la figura 6a.

rio puede llevar a cabo tareas complejas a un coste moderado, sin precisar el concurso de un computador central. Sin embargo, las estaciones dedicadas al procesamiento de imágenes no suelen aparecer autónomas *(stand-alone)*, sino conectadas por medio de una red local como la Ethernet. De este modo, los usuarios pueden compartir recursos comunes a todos los usuarios de la red, como son los dispositivos de almacenamiento de alta capacidad *(file servers)*, así como dispositivos de salida comunes. Ejemplos de este tipo de estaciones lo constituyen la estación SUN y APOLLO.

• *Computadores personales.* Los sistemas de procesamiento de imágenes basados en computadores personales son adecuados en aquellas aplicaciones en donde el bajo coste de la instalación sea un punto fundamental. Presentan la gran ventaja de disponer de un número muy amplio de tarjetas de adquisición de imágenes y de software asociado. Como ejemplos de este tipo de tarjetas cabe citar las de DATA TRANSLATION y las de PC-VISION.

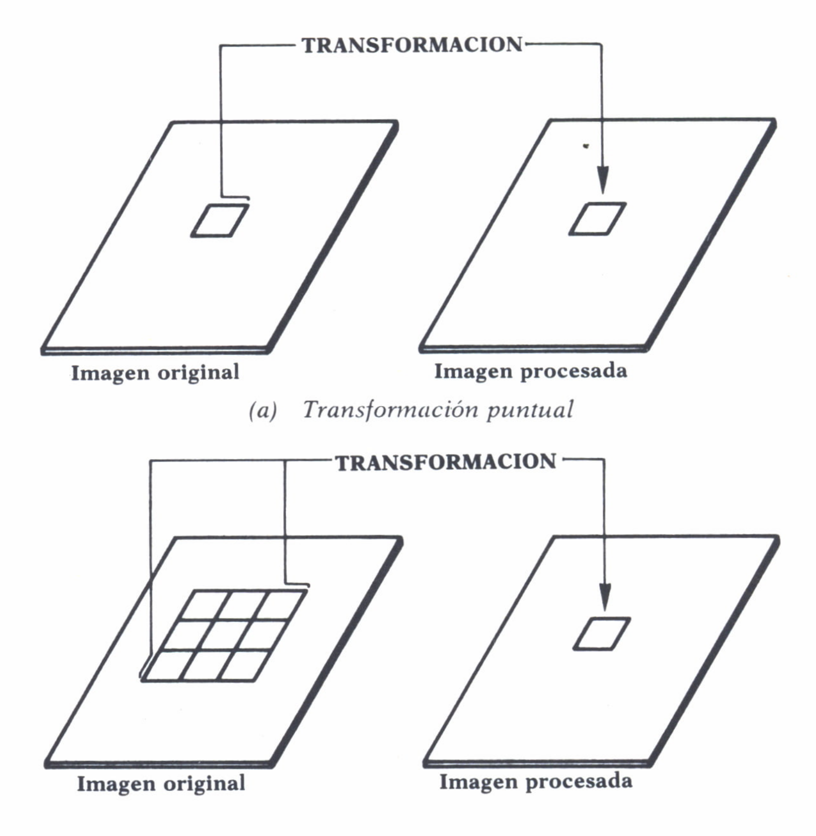


Características de los algoritmos para procesamiento de imágenes

Los algoritmos para procesamiento de imágenes deben presentar una característica común a todos ellos, como es su naturaleza no excesivamente compleja, puesto que deben ser aplicados a millones de datos. Como anteriormente se indicó, los algoritmos para procesamiento de imágenes pueden ser divididos en dos grandes grupos (figura 8):

• *Algoritmos de procesamiento puntual.* En este tipo de algoritmos, las operaciones se llevan a cabo sobre cada pixel individualizado, sin considerar el valor de los pixels próximos. Como ejemplos cabe citar las operaciones aritméticas y booleanas entre imágenes, las transformaciones radiométricas y las operaciones estadísticas. Las transformaciones radiométricas modifican los niveles de gris de una imagen por medio de la aplicación de funciones lineales y no lineales (incluyendo funciones aleatorias), dentro del rango de precisión del hardware disponible.

Constituyen la base de una gran parte de las operaciones de procesamiento, incluyendo la modificación del contraste. Las operaciones radiométricas se llevan a cabo utilizando las denominadas tablas de consulta (look-up tables: LUT), que se encuentran situadas entre la memoria de refresco y el monitor. Las tablas de consulta se encuentran implementadas en memoria RAM de acceso rápido (menor de 100 nseg.), empleándose la intensidad de los pixels de la memoria de refresco como una dirección en la tabla, proporcionando un valor de salida que se presenta en el monitor.



(a) Transformación puntual

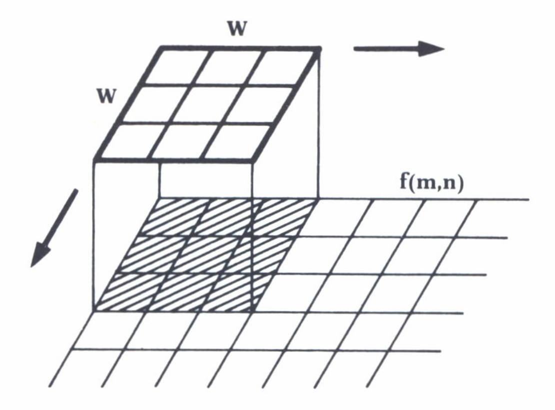
(b) Transformación regional

Fig. 8. Los dos tipos generales de procesamiento de imágenes.

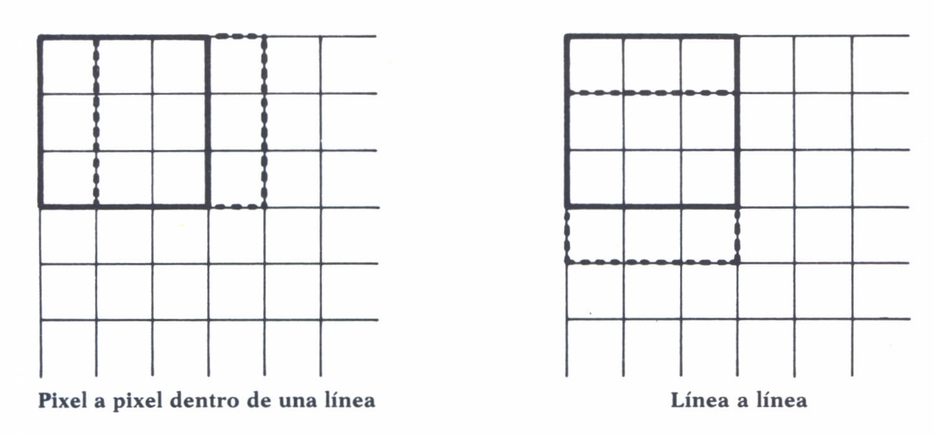
Este proceso se realiza a frecuencia de vídeo, permitiendo que los valores almacenados en la memoria de refresco puedan ser transformados en otros valores y representados en la pantalla de acuerdo con la función que se encuentre cargada en la tabla. Así, por ejemplo, la escala de gris puede ser invertida, o efectuar una transformación radiométrica logarítmica con objeto de aumentar el contraste de las zonas.

• *Algoritmos de procesamiento espacial.* En este tipo de algoritmos, la imagen resultante de las operaciones llevadas a cabo es función tanto de las intensidades de los pixels de entrada como de la de sus vecinos.

Los algoritmos de procesamiento puntual son de fácil implementación en los computadores convencionales, mediante la utilización de operaciones de entrada/salida. Sin embargo, los algoritmos de procesamiento espacial requieren un mayor grado de complejidad que los anteriormente señalados. Como ejemplos de operaciones de procesamiento espacial se encuentran las operaciones de convolución, en donde el resultado se obtie-



(a) Proyección de una ventana sobre una imagen que está siendo procesada.



(b) Cálculo de ventanas consecutivas.

Fig. 9. Ejemplos de operaciones de convolución.

ne multiplicando cada pixel de la imagen original por un núcleo o ventana de N × N pixels (figura 9). En la práctica, se utilizan núcleos de 3 × 3 o de 5 × 5 en las convoluciones en el dominio espacial, con objeto de que el tiempo necesario para llevar a cabo este tipo de operaciones no sea excesivamente grande.

Por otro lado, y por razones de coste, las operaciones de procesamiento espacial se realizan de forma iterativa, procesando un punto del núcleo o ventana por trama, con lo que se consigue un tiempo de ejecución próximo al tiempo real.

Un aspecto importante a tener en cuenta a la hora de evaluar los algoritmos para procesamiento de imágenes es el de su tiempo de ejecución. Podemos considerar tres tipos de intervalos temporales:

a) *Tiempo real.* Se considera que un algoritmo para procesamiento de imágenes se ejecuta en tiempo real si dicho tiempo es comparable al necesario para efectuar el refresco de las memorias de imagen, de tal manera que el usuario apenas aprecie que ha transcurrido un tiempo desde que lanzó la orden de ejecución y el momento en que se presenta el resultado.

b) *Próximo al tiempo real.* Se considera que un algoritmo para procesamiento de imágenes opera en un tiempo próximo al tiempo real si el tiempo de ejecución es del orden de 3 segundos. Como ejemplo de este tipo de algoritmos cabe señalar las convoluciones espaciales.

c) *No tiempo real.* Se considera que un algoritmo para procesamiento de imágenes opera en un tiempo muy superior al tiempo real si el tiempo de ejecución está por encima de los 3 segundos anteriormente señalados. Este tipo de algoritmos que suponen un tiempo elevado de ejecución se suelen introducir en una cola de procesos.



SISTEMAS DE PROCESAMIENTO DE IMAGENES BASADOS EN UN COMPUTADOR PERSONAL

Durante los últimos cinco años, los sistemas de procesamiento de imágenes basados en ordenadores personales han comenzado a jugar un papel importante en algunas áreas de aplicación, habiendo experimentado durante los últimos dos años un rápido crecimiento la tecnología de este tipo de sistemas. El progresivo descenso de los costes tanto del hardware como del software convierten al ordenador personal en una herramienta muy adecuada cuando el coste reducido del sistema es un requerimiento básico.

Una configuración típica de un sistema de procesamiento de imágenes basado en un ordenador personal IBM PC-AT con 512 Kbytes de memoria principal, 20 Mbytes de memoria en disco, requeriría el siguiente hardware adicional:

— Tarjeta de adquisición de imágenes.
— Cámara de vídeo y unidad de visualización.
— Un ratón conectado vía serie.

Opcionalmente puede añadirse el siguiente hardware:

— Tarjeta coprocesadora conectada a la tarjeta de adquisición de imágenes.
— Tarjeta adaptadora a un bus externo, multibús, AT, etc.
— Tarjeta de conexión a una red local: Ethernet, PC-NET, etc.
— Tarjeta de interface a un disco óptico. etc.

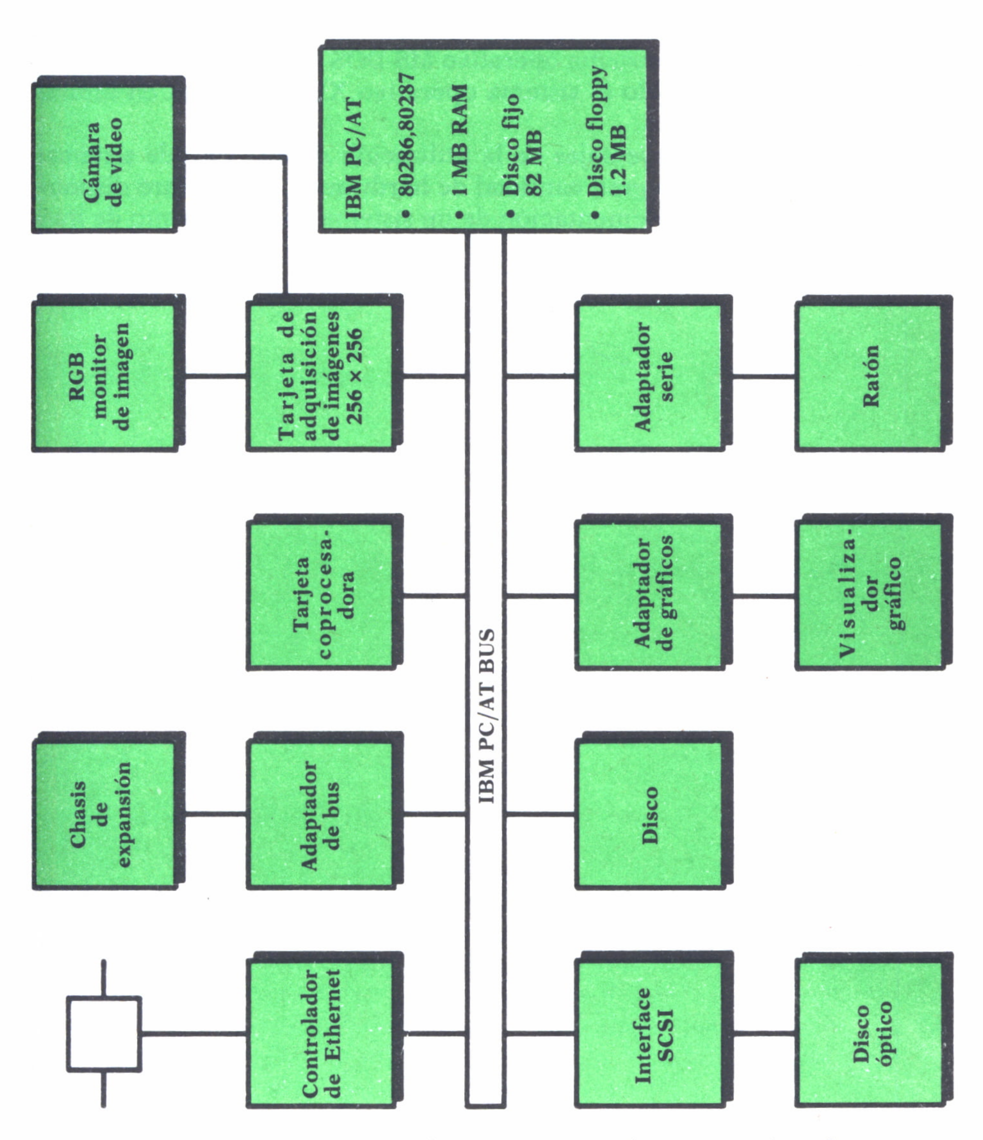


Fig. 10. Estructura de un sistema de procesamiento de imágenes basado en un ordenador personal.

En la figura 10 se presenta la estructura de un sistema de tratamiento de imágenes basado en un ordenador personal. En cuanto al software necesario, cabe señalar que el mayor inconveniente que presenta este tipo de configuraciones es la limitada capacidad de direccionamiento de memoria de la versión del sistema operativo MS-DOS 3.2. Este problema puede solucionarse utilizando el sistema operativo XENIX o bien en futuras versiones del MS-DOS.

Una de las mayores ventajas de la utilización de este tipo de sistemas reside en el hecho de la gran cantidad de hardware y de software disponibles que facilitan la implementación de un sistema de tratamiento de imágenes para un coste reducido.

APENDICE 1

ALGORITMO DE LA TRANSFORMADA RAPIDA DE FOURIER

Consideremos en primer lugar la expresión de la TDF:

$$F(u) = 1/N \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N] = 1/N \sum_{x=0}^{N-1} f(x)\, W_N^{ux} \qquad (A.1)$$

siendo $W_N = \exp[-j2\pi/N]$, N = 2n y N = 2M. Seguidamente, introduciremos un nuevo índice K en el sumatorio, relacionado con x mediante las siguientes relaciones:

x = 2K para x par
x = 2K + 1 para x impar

con lo que la expresión (A.1) puede escribirse como:

$$F(u) = 1/2M \sum f(k)\, W_{2M}^{uk} = 1/2\{1/M \sum f(2K)\, W_{2M}^{u(2k)} +$$
$$1/M \sum f(2K + 1)\, W_{2M}^{(2k+1)}\} = 1/2\{F_{par}(u) + F_{impar}(u)\, W_{2M}^{u}\} \qquad (A.2)$$

En definitiva, se ha reducido el cálculo de una transformada de N puntos a dos transformadas de (N/2) puntos. Así, por ejemplo, en el caso de una función con 8 puntos {f(0), f(1), ..., f(7)}, habría que considerar por un lado las muestras pares {f(0), f(2), f(4), f(6)} y por otro las impares {f(1), f(3), f(5), f(7)}. Cada una de las transformadas de 4 puntos se calcula como dos transformadas de 2 puntos. Así, para calcular la FFT de {f(0), f(2), f(4), f(6)} hay que dividir este conjunto en dos: la parte par {f(0), f(4)} y la parte impar {f(2), f(6)}. Análogamente se haría en el caso del segundo conjunto de las cuatro muestras impares: {f(1), f(3), f(5), f(7)}, que se dividiría en {f(1), f(5)} y {f(3), f(7)}. Así, pues, el algoritmo operaría según el esquema de la figura 1, de arriba a abajo:

Seguidamente se presenta una subrutina FORTRAN, propuesta por Cooley y Tuckey en 1969 que permite calcular la FFT mediante el método descrito anteriormente. Los parámetros de la subrutina son dos. F es un

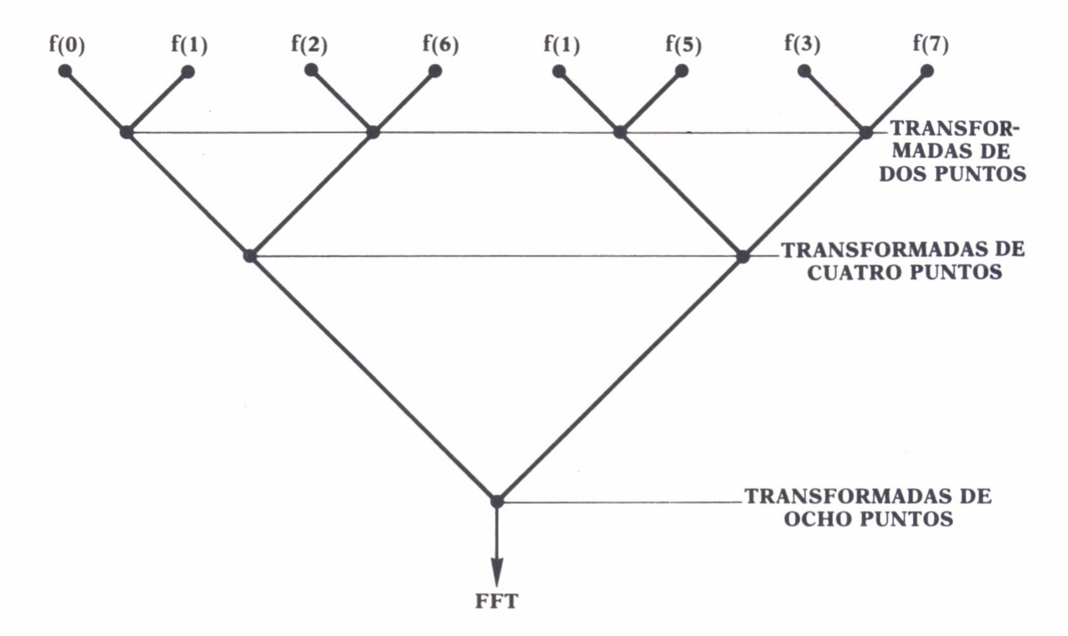


Fig. 1. Ordenación de los datos de la función {f(0), f(1), ..., f(7)} para el cálculo de la FFT.

array cuya transformada se desea calcular y LN = n, siendo N = 2n la dimensión del array considerado. Como resultado de los cálculos esta subrutina devuelve en F el valor de la transformada de Fourier. Obsérvese que F ha sido dimensionado en la subrutina como un array de números complejos, luego si partimos de un array F correspondiente a una función real, la parte imaginaria de F debe ser cero antes de llamar a esta subrutina.

```
      SUBROUTINE FFT(F,LN)
      COMPLEX F(1024),U,W,T,CMPLX
      PI=3.141593
      N=2**LN
      NV2=N/2
      NM1=N-1
      J=1
      DO 3 I=1,NM1
      IF(I.GE.J) GO TO 1
      T=F(J)
      F(J)=F(I)
      F(I)=T
    1 K=NV2
    2 IF(K.GE.J) GO TO 3
      J=J-K
```

```
      K=K/2
      GO TO 2
3     J=J+K
      DO 5 L=1,LN
      LE=2**L
      LE1=LE/2
      U=(1.0,0.0)
      W=CMPLX(COS(PI/LE1),-SIN(PI/LE1))
      DO 5 J=1,LE1
      DO 4 I=J,N,LE
      IP=I+LE1
      T=F(IP)*U
      F(IP)=F(I)+T
4     F(I)=F(I)+T
5     U=U*W
      DO 6 I=1,N
6     F(I)=F(I)/FLOAT(N)
      RETURN
      END
```

APENDICE 2

PROGRAMAS DE EXPERIMENTACION

El objetivo de los programas que se presentan en este apéndice consiste en proporcionar una herramienta sencilla, flexible y fácil de utilizar por aquellos usuarios de ordenadores personales. Los programas se han escrito en Fortran 77, requiriendo para su utilización una impresora de caracteres con objeto de poder imprimir los resultados.

Las imágenes utilizadas tienen un tamaño de 64 × 64 pixels, almacenando cada pixel como un valor complejo en punto flotante. Este tamaño de imágenes es muy apropiado para su presentación tanto en un monitor alfanumérico como para su impresión en una impresora de líneas, proporcionando suficiente resolución para el fin didáctico a que se dirige. Este formato de datos complejos se ha efectuado con objeto de facilitar el análisis de Fourier. La precisión de punto flotante asegura la no introducción de errores de redondeo. Por otro lado, el cálculo de la transformada de Fourier se ha efectuado de tal manera que la frecuencia cero corresponde al centro de la imagen presentada. La presentación en la pantalla se ha hecho utilizando una sola línea de caracteres para producir ocho niveles de seudogris.

En el programa se han definido tres buffers de imágenes de 64 × 64 pixels, lo cual permite la realización de operaciones de suma, escalado, transformación, etc. La utilización de este programa de entrenamiento es sencilla, puesto que se basa en la elección entre una lista de comandos predefinidos acompañados de uno o dos argumentos correspondientes a los «buffers» de imagen. El resto de los parámetros se introducen en respuesta a las peticiones efectuadas por el programa, asignándose valores por defecto en el caso de no especificar dichos parámetros.

Asimismo, el programa permite el diagnóstico de errores de sintaxis en los comandos, devolviendo control al usuario para su nueva introducción. Los errores producidos como consecuencia de la introducción de un pa-

rámetro erróneo implican la necesidad de introducir de nuevo el comando. En la figura 1 se presenta una imagen de barras y en la 2 el módulo de su transformada de Fourier. En la 3 se presenta una imagen constituida por un cuadrado de niveles de gris y en la 4 el módulo de su transformada de Fourier.

```
      PROGRAM AUTOMATA
C
C     Este programa es el  que maneja  por  medio
C     de  una  autómata  las  operaciones que  se
C     pueden efectuar en el procesamiento de imagenes.
C     El autómata maneja una tabla multiple que en
C     ayuda de un scannner se encarga de decodificar
C     una linea de comandos entregada a él.
C
      COMMON /BUFFER/ A,B,C
      COMPLEX A(4096),B(4096),C(4096)
      COMPLEX AA(64,64),BB(64,64),CC(64,64)
      EQUIVALENCE (A,AA),(B,BB),(C,CC)

      CHARACTER*1   COMANDO(13,30),LINEA(80),MANDATO(13)
      CHARACTER*80  LIN
      CHARACTER*220 PAL
      CHARACTER*22  PALABRAS(10)
      CHARACTER*13  NOMBRES(30),TOKEN,NOMBRE
      EQUIVALENCE   (NOMBRES,COMANDO),(TOKEN,MANDATO)
      EQUIVALENCE   (LIN,LINEA),(PALABRAS,PAL)
      COMMON /COMMAND/LIN,MANDATO
      COMMON /COMMAND1/LONGITUD,INDICE
      INTEGER DIRECCION(30)
C      direccion que se debe ejecutar despues del comando
      INTEGER REGRESO(30)
C      direccion de retorno una vez que se ejecuta
      INTEGER SUMA,RESTA,MULTIPLICACION,DIVISION,DIRECTA,INVERSA
      INTEGER ESCRIBIR,LEER,OPERACION,SALIR,CARACTER
      INTEGER REALOIMAG,CONJU,RECOPOL,POLOREC,NEG
      INTEGER LOGAE,LOGA10,REALO,IMAGO
      INTEGER CERO,CAMBIO,CUADRADO,ORGO,DFFT,INVFFT

C      Datas para las constantes que se ocuparan en el programa.

      DATA SUMA,RESTA,MULTIPLICACION,DIVISION/1,2,3,4/
      DATA ESCRIBIR,LEER/1,2/
      DATA SALIR/1/
      DATA REALOIMAG,CONJU,RECOPOL,POLOREC,
     *NEG,LOGAE,LOGA10,REALO,IMAGO/5,6,7,8,9,10,11,12,13/
      DATA CERO,CAMBIO,CUADRADO,ORGO,DFFT,INVFFT/14,15,16,17,18,19/


C     Máximo número de comandos que tiene la tabla
      DATA MAXIND /10/
C     ***********************************************************

C     Comandos que son aceptados.
      DATA NOMBRES/'DISCO            ',
     *            'SUMO             ',
     *            'RESTO            ',
```

```
     *              'RESTAR         ',
     *              'DIVIDIR        ',
     *              'DIVIDO         ',
     *              'MULTIPLICO     ',
     *              'MULTIPLICAR    ',
     *              'NEGAR          ',
     *              'SALIRSE        ',
     *              'SUME           ',
     *              'SUMAR          ',
     *              'RESTE          ',
     *              'DIVIDE         ',
     *              'MULTIPLIQUE    ',
     *              'NOT            ',
     *              'NIEGUE         ',
     *              'TERMINAR       ',
     *              'ACABAR         ',
     *              'EXIT           ',
     *              'END            ',
     *              'LOGOUT         ',
     *              'CAMBIE         ',
     *              'TRASLADE       ',
     *              'TRASLADO       ',
     *              'CAMBIAR        ',
     *              'LOGARITMO      ',
     *              'CERO           ',
     *              'CLEAR          ',
     *              'SQRT           '/
C
C     Palabras que se utilizan en una linea de comandos.
      DATA PALABRAS/
     *'BASE 10 REAL-IMAG R-I ',
     *'REAL IMAGINARIO CONJUG',
     *'ADO REC-POL RECTANGULA',
     *'R POLAR RP PR 2 ORIGEN',
     *' ORG DER-IZQ IZQUIERDA',
     *' DERECHA TRASLADE     ',
     *'       NEGAR PARTE BU',
     *'FFER  LEER ESCRIBIR   ',
     *'                      ',
     *'                      '/

C     Direcciones del regreso.
      DATA REGRESO/24,24,24,24,24,24,24,
     *24,24,24,24,24,24,24,24,24,24,24,
     *24,24,24,24,24,24,24,24,24,24,24,24/
C     Direcciones donde voy despues de decodificar el automata.
      DATA DIRECCION/
     *1,2,2,2,2,2,2,2,13,1,3,
     *3,4,4,5,13,13,1,1,1,1,
     *1,9,9,9,9,14,16,18,20/


C     COMIENZO EL PROCESO CON EL AUTOMATA.

30    CONTINUE      .

      CALL CLRSCR
C     ************************ PRUEBA ************************
      DO 15 K=1,3
```

```
      CALL PATRON(K,A)
      DO 13 I=0,11
        DO 14 J=1,64
           PRINT *,REAL(A(I*64+J))
14      CONTINUE
        PRINT *
13    CONTINUE
15    CONTINUE
      PRINT *,'FIN'
      STOP
C     ***********************************************************
      CALL LEERLINE
      INDICE=1
      CALL SCANNER
      JUMP=1
      J=1
20    CONTINUE
      LUGAR=1
      K=ICHAR(COMANDO(1,J))
      L=ICHAR(MANDATO(LUGAR))
      IF ( K.EQ.L) THEN
C     Veremos si este es el comando que se invoco
            I=2
10          CONTINUE
            K=ICHAR(MANDATO(I))
            L=ICHAR(COMANDO(I,J))
            IF (K.EQ.L) LUGAR=LUGAR+1
            I=I+1
            IF (LUGAR.EQ.LONGITUD) JUMP=DIRECCION(J)
            IF (LUGAR.EQ.LONGITUD) GO TO 40
            IF (I.LE.LONGITUD+1) GO TO 10
            ENDIF
      J=J+1
      IF (J.LE.MAXIND) GO TO 20
      GO TO 100
40    CONTINUE
C
C     ********************************************************
C     Comienzo el procesamiento del comando dado.
C
      IF ((JUMP.LT.1).AND.(JUMP.GT.24))GO TO 100
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,
     *1200,1300,1400,1500,1600,1700,1800,1900,2000,2100,
     *2200,2300,30 ),JUMP
C*************************************************************
C     Aqui nunca podre regresar de la salida del automata.  *
C                                                            *
C                                                            *
C*************************************************************

100   CONTINUE

      OPERACION=5
      CALL FILES(OPERACION,NOMBRE,A)
      PRINT *,LIN
      PRINT *
      GO TO 30
C     *******************************************************

200   CONTINUE
300   CONTINUE
```

```
400   CONTINUE
500   CONTINUE
C     Los comandos invocados son 200,300,400 y 500 para suma,
C     resta,multiplicación y división.
      CALL SCANNER
      CARACTER=ICHAR(MANDATO(1))
D     PRINT *,MANDATO(1),TOKEN,INDICE
D     PAUSE 'ESTOY EN 200 AL 500'
      IF (CARACTER.EQ.65) THEN
C     Era el buffer A.
              CALL SCANNER
D             PRINT *,MANDATO(1),INDICE,TOKEN,ichar(mandato(1))
D             PAUSE 'INGRESO AL TIPO DE OPERACION'
              OPERACION=SALIR
              CARACTER=ICHAR(MANDATO(1))
              IF (CARACTER.EQ.43) OPERACION=SUMA
              IF (CARACTER.EQ.45) OPERACION=RESTA
              IF (CARACTER.EQ.42) OPERACION=MULTIPLICACION
              IF (CARACTER.EQ.58) OPERACION=DIVISION
              IF (CARACTER.EQ.47) OPERACION=DIVISION
              IF (OPERACION.EQ.SALIR) GO TO 100
C     Era un signo de operacion válido.
D     PRINT *,INIDICE
              CALL SCANNER
              CARACTER=ICHAR(MANDATO(1))
D             PRINT *,MANDATO(1),INDICE,TOKEN,OPERACION
D             PAUSE 'DECODIFIQUE LA OPERACION VEREMOS DONDE VAMOS'
              IF (CARACTER.EQ.65) CALL FUNCION(OPERACION,A,A)
              IF (CARACTER.EQ.66) CALL FUNCION(OPERACION,A,B)
              IF (CARACTER.EQ.67) CALL FUNCION(OPERACION,A,C)
              IF ((CARACTER.NE.65).OR.(CARACTER.NE.66).OR.
     *          (CARACTER.NE.67)) GO TO 100

              JUMP=REGRESO(JUMP)
D             PRINT *,JUMP
D             PAUSE 'VEREMOS DONDE SALTO'
              GO TO 40
      ENDIF
      IF (CARACTER.EQ.66) THEN
C     Era el buffer B.
              CALL SCANNER
              OPERACION=SALIR
              CARACTER=ICHAR(MANDATO(1))
              IF (CARACTER.EQ.43) OPERACION=SUMA
              IF (CARACTER.EQ.45) OPERACION=RESTA
              IF (CARACTER.EQ.42) OPERACION=MULTIPLICACION
              IF (CARACTER.EQ.58) OPERACION=DIVISION
              IF (CARACTER.EQ.47) OPERACION=DIVISION
              IF (OPERACION.EQ.SALIR) GO TO 100

C     Era un signo de operacion válido.
              CALL SCANNER
              CARACTER=ICHAR(MANDATO(1))
              IF (CARACTER.EQ.65) CALL FUNCION(OPERACION,B,A)
              IF (CARACTER.EQ.66) CALL FUNCION(OPERACION,B,B)
              IF (CARACTER.EQ.67) CALL FUNCION(OPERACION,B,C)
              IF ((CARACTER.NE.65).OR.(CARACTER.NE.66).OR.
     *          (CARACTER.NE.67)) GO TO 100

              JUMP=REGRESO(JUMP)
              GO TO 40
```

```
      ENDIF
      IF (CARACTER.EQ.67) THEN
C     Era el buffer B.
              CALL SCANNER
              OPERACION=SALIR
              CARACTER=ICHAR(MANDATO(1))
              IF (CARACTER.EQ.43) OPERACION=SUMA
              IF (CARACTER.EQ.45) OPERACION=RESTA
              IF (CARACTER.EQ.42) OPERACION=MULTIPLICACION
              IF (CARACTER.EQ.58) OPERACION=DIVISION
              IF (CARACTER.EQ.47) OPERACION=DIVISION
              IF (OPERACION.EQ.SALIR) GO TO 100
C     Era un signo de operacion válido.
              CALL SCANNER
              CARACTER=ICHAR(MANDATO(1))
              IF (CARACTER.EQ.65) CALL FUNCION(OPERACION,C,A)
              IF (CARACTER.EQ.66) CALL FUNCION(OPERACION,C,B)
              IF (CARACTER.EQ.67) CALL FUNCION(OPERACION,C,C)
              IF ((CARACTER.NE.65).OR.(CARACTER.NE.66).OR.
     *          (CARACTER.NE.67)) GO TO 100

              JUMP=REGRESO(JUMP)
              GO TO 40
      ENDIF
      GO TO 100
C     ***************************************************
C
600   CONTINUE
C     Comando para leer en buffers.
      CALL SCANNER
      OPERACION=LEER
      IF (LONGITUD.GT.1) GO TO 100
      IF (CARACTER.EQ.65) THEN
              NOMBRE='BUFFERA.DAT'
              CALL FILES(OPERACION,NOMBRE,A)
              JUMP=REGRESO(JUMP)
              GO TO 40
              ENDIF
      IF (CARACTER.EQ.66) THEN
              NOMBRE='BUFFERB.DAT'
              CALL FILES(OPERACION,NOMBRE,B)
              JUMP=REGRESO(JUMP)
              GO TO 40
              ENDIF
      IF (CARACTER.EQ.67) THEN
              NOMBRE='BUFFERC.DAT'
              CALL FILES(OPERACION,NOMBRE,C)
              JUMP=REGRESO(JUMP)
              GO TO 40
              ENDIF
C     Error en lectura de buffers.
      GO TO 100
C
C     Comando para escribir los buffers en disco.
700   CONTINUE
      CALL SCANNER
      OPERACION=ESCRIBIR
      IF (LONGITUD.GT.1) GO TO 100
      IF (CARACTER.EQ.65) THEN
              NOMBRE='BUFFERA.DAT'
```

```
                CALL FILES(OPERACION,NOMBRE,A)
                JUMP=REGRESO(JUMP)
                GO TO 40
                ENDIF
      IF (CARACTER.EQ.66) THEN
                NOMBRE='BUFFERB.DAT'
                CALL FILES(OPERACION,NOMBRE,B)
                JUMP=REGRESO(JUMP)
                GO TO 40
                ENDIF
      IF (CARACTER.EQ.67) THEN
                NOMBRE='BUFFERC.DAT'
                CALL FILES(OPERACION,NOMBRE,C)
                JUMP=REGRESO(JUMP)
                GO TO 40
                ENDIF
C
C     A continuación estan todas las funciones que es posible
C     hacer con los buffers, SUMA, RESTA, DIVISION y MULTIPLICACION.
800   CONTINUE
      CALL SCANNER
      IF (CARACTER.EQ.65) THEN
                 CALL SCANNER
                 OPERACION=SALIR
                 IF (CARACTER.EQ.43) OPERACION=SUMA
                 IF (CARACTER.EQ.45) OPERACION=RESTA
                 IF (CARACTER.EQ.42) OPERACION=MULTIPLICACION
                 IF (CARACTER.EQ.58) OPERACION=DIVISION
                 IF (CARACTER.EQ.47) OPERACION=DIVISION
                 IF (OPERACION.EQ.SALIR) GO TO 100
                 CALL SCANNER
                 IF (CARACTER.EQ.65) CALL FUNCION(OPERACION,A,A)
                 IF (CARACTER.EQ.66) CALL FUNCION(OPERACION,A,B)
                 IF (CARACTER.EQ.67) CALL FUNCION(OPERACION,A,C)
                 IF ((CARACTER.NE.65).OR.(CARACTER.NE.66).OR.
     *            (CARACTER.NE.67)) GO TO 100
                 JUMP=REGRESO(JUMP)
                 GO TO 40
      ENDIF
      IF (CARACTER.EQ.66) THEN
                 CALL SCANNER
                 OPERACION=SALIR
                 IF (CARACTER.EQ.43) OPERACION=SUMA
                 IF (CARACTER.EQ.45) OPERACION=RESTA
                 IF (CARACTER.EQ.42) OPERACION=MULTIPLICACION
                 IF (CARACTER.EQ.58) OPERACION=DIVISION
                 IF (CARACTER.EQ.47) OPERACION=DIVISION
                 IF (OPERACION.EQ.SALIR) GO TO 100
                 CALL SCANNER
                 IF (CARACTER.EQ.65) CALL FUNCION(OPERACION,B,A)
                 IF (CARACTER.EQ.66) CALL FUNCION(OPERACION,B,B)
                 IF (CARACTER.EQ.67) CALL FUNCION(OPERACION,B,C)
                 IF ((CARACTER.NE.65).OR.(CARACTER.NE.66).OR.
     *            (CARACTER.NE.67)) GO TO 100
                 JUMP=REGRESO(JUMP)
                 GO TO 40
      ENDIF
      IF (CARACTER.EQ.67) THEN
                 CALL SCANNER
                 OPERACION=SALIR
                 IF (CARACTER.EQ.43) OPERACION=SUMA
```

```
                IF (CARACTER.EQ.45) OPERACION=RESTA
                IF (CARACTER.EQ.42) OPERACION=MULTIPLICACION
                IF (CARACTER.EQ.58) OPERACION=DIVISION
                IF (CARACTER.EQ.47) OPERACION=DIVISION
                IF (OPERACION.EQ.SALIR) GO TO 100
                CALL SCANNER
                IF (CARACTER.EQ.65) CALL FUNCION(OPERACION,C,A)
                IF (CARACTER.EQ.66) CALL FUNCION(OPERACION,C,B)
                IF (CARACTER.EQ.67) CALL FUNCION(OPERACION,C,C)
                IF ((CARACTER.NE.65).OR.(CARACTER.NE.66).OR.
     *                (CARACTER.NE.67)) GO TO 100
                JUMP=REGRESO(JUMP)
                GO TO 40
      ENDIF
C
C
C     Procedimiento para obtener filtro en imagen o bien FFT directa

C     inversa. Ademas cambios varios en una matriz.

C     Intercambio parte REAL por IMAGINARIA
900   CONTINUE
      OPERACION=REALOIMAG
      CALL SCANNER
      GO TO 1

C     Procedimiento para obtener el conjugado de una matriz.
1000   CONTINUE
      OPERACION=CONJU
      GO TO 1
C
C     Procedimiento para obtener el cambio de rectangular a polar.
1100  CONTINUE
      OPERACION=RECOPOL
      GO TO 1
C
C     Procedimiento para convertir de polar a rectangular.
1200  CONTINUE
      OPERACION=POLOREC
      GO TO 1

C
C     Procedimiento para negar una matriz.
1300  CONTINUE
      OPERACION=NEG
      GO TO 1

C
C     Procedimiento para sacar el logarítmo en base E de una matriz
C     Procedimiento para obtener el logarítmo en base 10 de una matr
      iz.
1400  CONTINUE
1500  CONTINUE
      CALL SCANNER
      K=INDEX(PAL,TOKEN)
      IF (K.EQ.1) THEN
                OPERACION=SALIR
                CALL SCANNER
                K=INDEX(PAL,TOKEN)
                IF (K.EQ.4) OPERACION=LOGAE
```

```
                IF (K.EQ.6) OPERACION=LOGA10
                IF(OPERACION.EQ.SALIR) GO TO 100
                GO TO 1
      ELSEIF(K.EQ.4) THEN
                OPERACION=LOGAE
                GO TO 1
      ELSEIF(K.EQ.6) THEN
                OPERACION=LOGA10
                GO TO 1
      ENDIF
      GO TO 100
C
C     Procedimiento para llevar a cero la parte REAL de una matriz.
C     o la parte IMAG
1600  CONTINUE
1700  CONTINUE
      CALL SCANNER
      K=INDEX(PAL,TOKEN)
      IF(K.EQ.150) THEN
                 CALL SCANNER
                 OPERACION=SALIR
                 K=INDEX(PAL,TOKEN)
                 IF(K.EQ.11) OPERACION=REALO
                 IF(K.EQ.14) OPERACION=IMAGO
                 IF(SALIR.EQ.OPERACION)GO TO 100
                 GO TO 1
      ELSEIF(K.EQ.11) THEN
                 OPERACION=REALO
                 GO TO 1
      ELSEIF(K.EQ.14) THEN
                 OPERACION=IMAGO
                 GO TO 1
      ENDIF
C
C     Procedimiento para borrar el buffer indicado.
1800  CONTINUE
      CALL SCANNER
      K=INDEX(PAL,TOKEN)
      IF(K.EQ.58) THEN
                OPERACION=CERO
                GO TO 1
      ELSEIF(K.EQ.153) THEN
                GO TO 1
      ELSEIF(K.EQ.1) THEN
                INDICE=INDICE-1
                GO TO 1
      ELSEIF(K.EQ.2) THEN
                INDICE=INDICE-1
                GO TO 1
      ELSEIF(K.EQ.39) THEN
                INDICE=INDICE-1
                GO TO 1
      ENDIF
      GO TO 100

C
C     Procedimiento para intercambiar parte derecha por
C     la izquierda en una imagen.
1900  CONTINUE
      OPERACION=CAMBIO
      GO TO 1
```

```
C
C     Procedimiento para sacar  el cuadrado de la magnitud.
2000  CONTINUE
      OPERACION=CUADRADO
      GO TO 1

C
C     Procedimiento para poner en cero el origen de la imagen.
2100  CONTINUE
      OPERACION=ORGO
      GO TO 1

C
C
C     Procedimiento para obtener la transformada
C     de FOURIER de la imagen
2200  CONTINUE
      OPERACION=DFFT
      GO TO 1

C
C     Procedimiento para obtener la tranformada inversa de FOURIER.
2300  CONTINUE
      OPERACION=INVFFT
      GO TO 1

C     Procedimiento para procesar todos los comandos indicados.
C
    1 CONTINUE
      CALL SCANNER
      IF (LONGITUD.GT.1) GO TO 100
      IF (CARACTER.EQ.65) CALL FUNCION(OPERACION,A,B)
      IF (CARACTER.EQ.66) CALL FUNCION(OPERACION,B,A)
      IF (CARACTER.EQ.67) CALL FUNCION(OPERACION,C,A)
      IF ((CARACTER.NE.65).OR.(CARACTER.NE.66).OR.
     * (CARACTER.NE.67)) THEN
                              GO TO 100
                              ELSE
                              JUMP=REGRESO(JUMP)
                              GO TO 40
                              ENDIF
      JUMP=1
      GO TO 40
      END
      END



      SUBROUTINE FUNCION(CODIGO,A,B)
C
C
C     ESTA SUBROUTINE EFECTUA LAS OPERACINES ARITMETICAS
C     SOBRE EL PLANO IMAGEN DE MATRICES A Y B AMBAS COMPLEJAS.
C
C     CODIGO=1
C        SE EFECTUA LA SUMA DE MATRICES A=A + B
C
C     CODIGO=2
C        SE EFECTUA LA RESTA DE MATRICES A=A - B
```

```
C
C       CODIGO=3
C          SE EFECTUA LA MULTIPLICACION DE MATRICES A=A * B
C
C       CODIGO=4
C          SE EFECTUA LA DIVISION DE MATRICES A=A / B
C
C       CODIGO=5
C          SE EFECTUA EL TRASLADO DE A --> B Y B --> A
C
C       CODIGO=6
C          SE EFECTUA EL TRASLADO DE PARTE REAL POR IMAGINARIA
C
C       CODIGO=7
C          SE EFECTUA EL CAMBIO DE MATRIZ POR EL CONJUGADO.
C
C       CODIGO=8
C          SE EFECTUA EL CAMBIO DE RECTANGULAR A POLAR.
C
C       CODIGO=9
C          SE EFECTUA EL CAMBIO DE POLAR A RECTANGULAR.
C
C       CODIGO=10
C          SE EFECTUA EL NEGADO DE LA MATRIZ.
C
C       CODIGO=11
C          SE EFECTUA EL LOGARITMO EN BASE E.
C
C       CODIGO=12
C          SE EFECTUA EL LOGARITMO EN BASE 10.
C
C       CODIGO=13
C          SE EFECTUA LA PUESTA A CERO DE LA PARTE REAL.
C
C       CODIGO=14
C          SE EFECTUA LA PUESTA A CERO DE LA PARTE IMAGINARIA
C
C       CODIGO=15
C          SE BORRA EL BUFFER.
C
C       CODIGO=16
C          SE INTERCAMBIA PARTE DERECHA POR IZQUIERDA EN LA IMAGEN.
C
C       CODIGO=17
C          SE SACA EL CUADRADO DE LA MAGNITUD.
C
C       CODIGO=18
C          SE PONE A CERO EL ORIGEN.
C
        COMMON /COMMAND/LINEA,COMANDO
        COMPLEX RESP(4096),AA(64,64),BB(64,64)
        COMPLEX A(4096),B(4096)
        EQUIVALENCE (RESP,AA,BB)
        INTEGER CODIGO
C
        IF(CODIGO.EQ.1) THEN
C       SUMA DE MATRICES COMPLEJAS
                             DO 10 I=1,4096,1
                             A(I)=A(I) + B(I)
10                           CONTINUE
```

```
      ELSEIF (CODIGO.EQ.2) THEN
C     RESTA DE MATRICES
                          DO 20 I=1,4096,1
                          A(I)=A(I) - B(I)
20                        CONTINUE
      ELSEIF (CODIGO.EQ.3) THEN
C     MULTIPLICACION DE MATRICES
                          DO 30 I=1,4096,1
                          A(I)=A(I) * B(I)
30                        CONTINUE
      ELSEIF (CODIGO.EQ.4) THEN
C     DIVISION DE MATRICES
                          DO 40 I=1,4096,1
                          A(I)=A(I) / B(I)
40                        CONTINUE
      ELSEIF (CODIGO.EQ.5) THEN
C     SE HACE EL INTERCAMBIO DE A --> B Y B --> A
                          DO 50 I=1,4096,1
                          RESP(I)=A(I)
                          A(I)=B(I)
                          B(I)=RESP(I)
50                        CONTINUE
                          RETURN
      ELSEIF (CODIGO.EQ.6) THEN
C     INTERCAMBIO PARTE REAL POR IMAGINARIA
      PRINT *,' Intercambio de parte REAL por IMAGINARIA '
      DO 101 I=1,4096
      PREAL=REAL(A(I))
      PIMAG=AIMAG(A(I))
      A(I)=CMPLX(PIMAG,PREAL)
101   CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.7) THEN
C     CAMBIO LA MATRIZ POR EL CONJUGADO.
200   PRINT *,' cambio matriz por el CONJUGADO'
      DO 201 I=1,4096
      A(I)=CONJG(A(I))
201   CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.8) THEN
C     CAMBIO DE RECTANGULAR A POLAR.
300   PRINT *,' Cambio de rectangular a polar '
      DO 301 I=1,4096
      VALOR=CABS(A(I))
      ANGULO=90.0*SIGN(1.0,AIMAG(A(I)))
      PREAL=REAL(A(I))
      PIMAG=AIMAG(A(I))
      IF (REAL(A(I)).NE.0.0) ANGULO=57.29578*ATAN2(PIMAG,PREAL)
      A(I)=CMPLX(VALOR,ANGULO)
301   CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.9) THEN
C     CAMBIO DE POLAR A RECTANGULAR.
400   PRINT *,' Convierto de polar a rectangular '
      DO 401 I=1,4096
      ANGULO=0.01745329*AIMAG(A(I))
      PREAL=COS(ANGULO)*REAL(A(I))
      PIMAG=SIN(ANGULO)*REAL(A(I))
      A(I)=CMPLX(PREAL,PIMAG)
401   CONTINUE
```

```
      RETURN
      ELSEIF (CODIGO.EQ.10) THEn
C     NEGADO DE LA MATRIZ.

500   PRINT *,' El negado de la matriz '
      DO 501 I=1,4096
      A(I)=-A(I)
501   CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.11) THEN
C     LOGARITMO EN BASE E
600   PRINT *,' Logarítmo en base E '
      DO 601 I=1,4096
      VALOR=REAL(A(I))
      IF (VALOR.EQ.0.0) VALOR=10.0E-20
      VALOR=ALOG(VALOR)
      A(I)=CMPLX(VALOR,0.0)
601   CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.12) THEN
C     LOGARITMO EN BASE 10.
700   PRINT *,' Logarítmo en base 10'
      DO 701 I=1,4096
      VALOR=REAL(A(I))
      IF (VALOR.EQ.0.0) VALOR=10.0E-20
      VALOR=2.30259*ALOG(VALOR)
      A(I)=CMPLX(VALOR,0.0)
701   CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.13) THEN
C     PONER A CERO LA PARTE REAL.
800   PRINT *,' Poner a cero la parte real'
      DO 801 I=1,4096
      VALOR=AIMAG(A(I))
      A(I)=CMPLX(0.0,VALOR)
801   CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.14) THEN
C     PONER A CERO LA PARTE IMAGINARIA.
900   PRINT *,' Poner a cero la parte imaginaria'
      DO 901 I=1,4096
      VALOR=AIMAG(A(I))
      A(I)=CMPLX(VALOR,0.0)
901   CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.15) THEN
C     BORRAR EL BUFFER.
1000  PRINT *,' Borro el buffer '
      DO 1001 I=1,4096
      A(I)=CMPLX(0.0,0.0)
1001  CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.16) THEN
C     INTERCAMBIO PARTE DERACHA POR IZQUIERDA DE LA IMAGEN.
2000  PRINT *,' Intercambio parte derecha',
     *' por parte izquierda en la imagen'
      DO 2002 I=1,4096
      RESP(I)=A(I)
2002  CONTINUE
      DO 2001 I=1,32
```

```
      II=65-I
      DO 2001 J=1,64
      BAK=AA(J,I)
      AA(J,I)=AA(J,II)
      AA(J,II)=BAK
2001  CONTINUE
      DO 2003 I=1,4096
      A(I)=RESP(I)
2003  CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.17) THEN
C     CUADRADO DE LA MAGNITUD.
3000  PRINT *,' Cuadrado de la magnitud '
      DO 3001 I=1,4096
      A(I)=CABS(A(I))**2
3001  CONTINUE
      RETURN
      ELSEIF (CODIGO.EQ.18) THEN
C     PONER A CERO EL ORIGEN DE LA IMAGEN.
4000  PRINT *,' Poner a cero el origen '
      A(1089)=(0.0,0.0)
      RETURN
      ENDIF
      END
      SUBROUTINE PATRON(PAT,A)

C     ESTA RUTINA GENERA EN LA MATRIZ 'A' UN PATRON DE TEST:

C
C        PAT = 1    ==> FRANJAS HORIZONTALES
C
C        PAT = 2    ==> FRANJAS VERTICALES
C
C        PAT = 3    ==> AJEDREZ

      INTEGER PAT,CONTA
      REAL BLANCO
      REAL NEGRO
      REAL VALOR,IMA
      COMPLEX A(4096)

      BLANCO = 10.0
      NEGRO = -10.0
      IF (PAT.EQ.1) THEN
C     PATRON DE FRANJAS HORIZONTALES
                   CONTA=0
                   VALOR=BLANCO
                   DO 10 I=1,4096,1
                     CONTA=CONTA+1
                     IF (CONTA.EQ.256) THEN
                         CONTA=0
                         VALOR = -VALOR
                     ENDIF
                     IMA=AIMAG(A(I))
                     A(I)=CMPLX(VALOR,IMA)
10                 CONTINUE
      ELSEIF (PAT.EQ.2) THEN
C     PATRON DE FRANJAS VERTICALES
                   CONTA=0
                   VALOR=BLANCO
```

```
                    DO 20 I=1,4096,1
                      CONTA=CONTA+1
                      IF (CONTA.EQ.4) THEN
                          CONTA=0
                          VALOR = -VALOR
                      ENDIF
                      IMA=AIMAG(A(I))
                      A(I)=CMPLX(VALOR,IMA)
20                  CONTINUE
      ELSEIF (PAT.EQ.3) THEN
C     PATRON DE AJEDREZ
                    DO 30 I=1,4096,1
                    IF ((I MOD 2).EQ.1) THEN
                      IF ((FILA MOD 2).EQ.1) THEN
                          VALOR = BLANCO
                      ELSE
                          VALOR = NEGRO
                      ENDIF
                    ELSEIF ((FILA MOD 2).EQ.1) THEN
                          VALOR = NEGRO
                      ELSE
                          VALOR = BLANCO
                    ENDIF
                    A(I)=CMPLX(VALOR,VALOR)
30                  CONTINUE
      ENDIF
      RETURN
      END
C     *****************************************************

      SUBROUTINE FILES(OPERACION,ARCHIVO,MATRIZ)
C
C     ESTA RUTINA HACE LA APERTURA, CERRADO, LECTURA
C     O ESCRITURA DE UN ARCHIVO. LA LECTURA O ESCRITURA
C     SE HACE POR MEDIO DE LA MATRIZ B.
C
C     OPERACION=1
C               ENTONCES SE ABRE UN ARCHIVO.
C
C     OPERACION=2
C               ENTONCES SE CIERRA UN ARCHIVO.
C
C     OPERACION=3
C               ENTONCES SE LEE UN ARCHIVO.
C
C     OPERACION=4
C               ENTONCES SE ESCRIBE UN ARCHIVO.
C
C     OPERACION=5
C               ENTONCES SE ESCRIBE UN MENSAJE
C               DE COMANDO MAL FORMADO.
C
C     LOS DATOS LEIDOS O HA ESCRIBIR SE DEBEN
C     ENTREGAR EN LA MATRIZ B DEL
C     BUFFER ARCH.
C
C     *******************************************************

      COMPLEX B(4096),BB(64,64),MATRIZ(4096)
      EQUIVALENCE (B,BB)
```

```
      CHARACTER*12 ARCHIVO
      INTEGER OPERACION

C
      IF (OPERACION.EQ.1) THEN
C     ABRO UN ARCHIVO.
                        OPEN(UNIT=4,FILE=ARCHIVO,
     *  ACCESS='DIRECT',RECL=512)
                        RETURN
                        ELSEIF (OPERACION.EQ.2) THEN
C     CERRAR UN ARCHIVO.
                        CLOSE(UNIT=4)
                        RETURN
                        ELSEIF (OPERACION.EQ.3) THEN
C     SE LEE UN ARCHIVO EN LA MATRIZ DEL BUFFER ARCH LLAMADA B.
                        DO 10 I=1,64,1
                        DO 10 J=1,64,1
                        READ(UNIT=4,REC=I)BB(I,J)
10                      CONTINUE
                        DO 11 I=1,4096
                        MATRIZ(I)=B(I)
11                      CONTINUE
                        RETURN
                        ELSEIF (OPERACION.EQ.4) THEN
C     ESCRIBO EL BUFFER ARCH EN EL DISCO.
                        DO 21 I=1,4096
                        B(I)=MATRIZ(I)
21                      CONTINUE
                        DO 20 I=1,64,1
                        DO 20 J=1,64,1
                        WRITE(UNIT=4,REC=I)BB(I,J)
20                      CONTINUE
                        RETURN
                        ELSEIF (OPERACION.EQ.5) THEN
C     ESCRIBO UNA PANTALLA DICIENDO QUE EL ERROR SE
C     DEBE A ILEGAL COMANDO.
                        CALL CLRSCR
                        PRINT *,' ERROR: el comando',
     *' que invoco es ilegal.'
                        RETURN
                        ENDIF

      RETURN
      END

      SUBROUTINE CLRSCR
      CHARACTER*1 FILA(10)
      CHARACTER*10 BORRAR
      EQUIVALENCE (FILA,BORRAR)
      DATA FILA/Z'5C',Z'1B',Z'5B',Z'32',Z'4A',
     * Z'00',Z'00',Z'00',Z'00',Z'00'/
      PRINT 100,BORRAR
100   FORMAT(A10)
      END
C     ****************************************

      SUBROUTINE GOTOXY(I,J,PONER)
      CHARACTER*1 FILA(9)
      CHARACTER*2 FILA1,FILA2
      CHARACTER*9 PONER
```

```
      DATA FILA(1),FILA(2),FILA(3)/Z'5C',Z'1B',Z'5B'/
      k=i- (i/10)*10 + ICHAR('0')
      FILA1=CHAR(I/10 + ICHAR('0'))//CHAR(K)
      K=J- (J/10)*10 + ICHAR('0')
      FILA2=CHAR(J/10 + ICHAR('0'))//CHAR(K)
      PONER=FILA(1)//FILA(2)//FILA(3)//FILA1//';'//FILA2//'H'
      RETURN
      END


      SUBROUTINE GOTO
      CHARACTER*1 FILA(11)
      CHARACTER*11 PONER
      EQUIVALENCE (FILA,PONER)
      DATA FILA/Z'5C',Z'1B',Z'5B',Z'30',Z'36',Z'3B',Z'31',Z'32',
     * Z'48',Z'00',Z'00'/
      PRINT 100,PONER
100   FORMAT(1h+,A9)
      END
      SUBROUTINE FRAME(UPPERLEFTX,UPPERLEFTY,LOWERRIGHTX,LOWERRIGHY)
C
C     PARAMETROS PARA MANEJAR EL CURSOR:
C                 UPPERLEFTX:
C                             EJE IZQUIERDO X
C
C                 UPPERLEFTY:
C                             EJE IZQUIERDO Y
C
C                 LOWERRIGHX:
C                             EJE DERECHO PARTE DE ABAJO EN X
C
C                 LOWERRIGHY:
C                             EJE DERECHO PARTE DE ABAJO EN Y
C
C
C     *****************************************************
      INTEGER     UPPERLEFTX,UPPERLEFTY
      CHARACTER*1  LEFTCORNER,RIGHTCORNEL,LINEA,MURALLA
      CHARACTER*9 PONER

      CALL GOTOXY(UPPERLEFTX,UPPERLEFTY,PONER)
      LINEA=1h_
      LEFTCORNEL=1H┌
      RIGHTCORNEL=1H┐
      MURALLA=1H|
      PRINT 100,LEFTCORNEL
      DO 10 I=UPPERLEFTX+1,LOWERRIGHTX-1,1
      PRINT 100,LINEA
10    CONTINUE
100   FORMAT(1H+,80A1)
      PRINT 100,RIGHTCORNEL
      DO 20 I=UPPERLEFTY+1,LOWERRIGHTY-1,1
      CALL GOTOXY(UPPERLEFTX,I,PONER)
      PRINT 100,MURALLA
      CALL GOTOXY(LOWERRIGHTX,I,PONER)
      PRINT 100,MURALLA
20    CONTINUE
      CALL GOTOXY(UPPERLEFTX,LOWERRIGHTY,PONER)
      LEFTCORNEL=1H└
      RIGHTCORNEL=1H┘
```

```
      PRINT 100,LEFTCORNEL
      DO 30 I=UPPERLEFT+1,LOWERRIGHTX-1,1
      PRINT 100,LINEA
30    CONTINUE
      PRINT 100,RIGHTCORNEL
      RETURN
      END

      SUBROUTINE LEERLINE
C
C     Esta rutina lee una linea de comando desde
C     el terminal TTY como máximo lee 80
C     caracteres en LINEA.
C
      COMMON /COMMAND/LIN
      CHARACTER*80 LIN
C
C
      READ(*,100)LIN
100   FORMAT(A80)
      END

      SUBROUTINE SCANNER
C
C     Esta subroutina hace el scanner de la linea
C     introducida por la TTY luego se procede a
C     efectuar el parse de la misma linea.
C
      COMMON /COMMAND/LINEA,COMANDO
      COMMON /COMMAND1/LONGITUD,INDICE
      CHARACTER*13 COMANDO
      CHARACTER*1  TOKEN(13),LINEA(80),CR,TAB,BLANCO
      INTEGER CARACTER
      EQUIVALENCE  (TOKEN,COMANDO)
      LOGICAL FLAG
      DATA CR,BLANCO,TAB/Z'0D',Z'20',Z'09'/
C
      FLAG=.FALSE.
      LONGITUD=0
      DO 10 I=1,13
10    TOKEN(I)=BLANCO
C     Ignoro los blancos al comienzo, blancos y tabs
      I=INDICE
      J=1
100   CONTINUE
      CARACTER=ICHAR(LINEA(I))
      IF ((CARACTER.EQ.32) .OR. (CARACTER.EQ.9)) THEN
                            I=I+1
                            GO TO 100
                            ENDIF
200   CONTINUE
      IF ((ICHAR(LINEA(I)).GE.97).AND.(ICHAR(LINEA(I)).LE.122))
      THEN
C     Convierto de minusculas a mayusculas y
C     pongo donde debe estar.
                            TOKEN(J)=CHAR( ICHAR( LINEA(I) ) - 32)
                            I=I+1
                            J=J+1
```

```
                        GO TO 200
                        ENDIF
C       Verifico si es mayúscula.
        IF ((ICHAR(LINEA(I)).GE.65).AND.(ICHAR(LINEA(I)).LE.90)) THEN
C       Era mayúscula la guardo en lugar seguro.
                        TOKEN(J)=LINEA(I)
                        I=I+1
                        J=J+1
                        GO TO 200
                        ENDIF
C       No era letra mayúscula ni minuscula
C       veremos si es un número.
        IF ((ICHAR(LINEA(I)).GE.48).AND.(ICHAR(LINEA(I)).LE.57)) THEN
C       Era un número lo guardo en lugar seguro.
                        TOKEN(J)=LINEA(I)
                        I=I+1
                        J=J+1
                        GO TO 200
                        ENDIF
C       Veremos que cosa extraña puede ser.
        IF (((ICHAR(LINEA(I)).EQ.61).OR.(ICHAR(LINEA(I)).EQ.42).OR.
     * (ICHAR(LINEA(I)).EQ.43).OR.(ICHAR(LINEA(I)).EQ.47).OR.
     * (ICHAR(LINEA(I)).EQ.58)
     *.OR.(ICHAR(LINEA(I)).EQ.45)) .AND.(J.EQ.1)) THEN
C       Es un signo *,/,:,-,+ o bien =.
                        TOKEN(J)=LINEA(I)
                        I=I+1
                        INDICE=I
                        LONGITUD=J
                        RETURN
                        ENDIF
C       Era un delimitador.
        INDICE=I
        LONGITUD=J-1
        RETURN
        END


        SUBROUTINE FILTROS(FILTRO)
C
C       ESTA SUBROUTINA CALCULA DIFERENTES TIPOS DE
C       FILTROS PARA EL MANEJO DE IMAGEN
C       EL VALOR DE LA VARIABLE FILTRO NOS INDICA
C       CUAL SERA EL FILTRO A GENERAR
C
C       FILTRO=1
C               ENTONCES SE CALCULA FILTRO BUTTERWORTH.
C
C       FILTRO=2
C               ENTONCES SE CALCULA EL FILTRO EXPONENCIAL.
C
C       FILTRO=3
C               ENTONCES SE CALCULA EL FILTRO TRAPEZOIDAL.
C
C       FILTRO=4
C               ENTONCES SE CALCULA EL FILTRO LAPLACIANO.
C
C       *******************************************************

        COMMON /BUFFER/ A
        COMPLEX A(4096),AB(64,64)
```

```
      EQUIVALENCE (A,AB)
      INTEGER*2 FILTRO
C     AQUI DEBE IR UN IF THEN ELSE.

      CALL CLRSCR
      CALL GOTOXY(2,10,PONER)
      WRITE(*,100)
100   FORMAT(1H+,'Deme el valor de los puntos de corte en ejes X e Y

      READ(*,200)CORTEX,CORTEY
200   FORMAT(2F10.0)
      IF (CORTEX.LE.0.0) CORTEX=1.0
      IF (CORTEY.LE.0.0) CORTEY=1.0
      CALL GOTOXY(2,10,PONER)
      WRITE(*,300)
300   FORMAT(1H+,'Deme el ancho en eje X de la transición :')
      READ(*,400)ANCHOX
400   FORMAT(F10.0)
      IF( ANCHOX.LE.0.0)ANCHOX=1.0
      CALL GOTOXY(2,10,PONER)
10    WRITE(*,500)
500   FORMAT(1H+,'Quiere filtro pasa alto (1) o filtro pasa bajos (2

      READ(*,600)TIPO
600   FORMAT(F10.0)
      IF ((TIPO.NE.1.0).OR.(TIPO.NE.2.0)) GO TO 10
      CALL GOTOXY(2,10,PONER)
      WRITE(*,700)
700   FORMAT(1H+,'Orden del filtro :')
      READ(*,800)ORDEN
800   FORMAT(F10.0)
      IF (ORDEN.LE.0.0) ORDEN=1.0
      DO 20 I=1,33
      DO 20 J=1,33
      I1=32+I
      I2=34-I
      J1=32+J
      J2=34-J
      DX=FLOAT(I-1)/CORTEX
      DY=FLOAT(J-1)/CORTEY
      D=CABS(CMPLX(DX,DY))
      IF (FILTRO.NE.3) THEN
                       IF(ORDEN.NE.1.0.AND.D.NE.0.0) D=D**ORDEN
                       X=0.0
                       IF (FILTRO.EQ.1) X=1.0/(1.0 + 0.412*D)
                       IF (FILTR.EQ.2 .AND. D .LT. 100) X=EXP(-0.347
                       *D)
                       IF ( FILTRO.EQ.4 ) X=D
                       ENDIF
      X=ORDEN
      IF(D.LE.1.0) GO TO 22
      D=(D-1.0)*CORTEX
      X=1.0 - ORDEN
      IF(D.GT.ANCHOX) GO TO 22
      X=(ANCHOX-D)/ANCHOX
      IF(TIPO.EQ.1.0) X=1.0 - X
      IF (ORDEN.NE.1.0)X=X**ORDEN
22    CONTINUE
      IF (D.EQ.0.0 .AND. FILTRO.NE.4) X=TIPO
```

```
C       LLENO AHORA LOS PIXEL'S

        IF (J1.EQ.65 ) GO TO 40
        IF( I1.EQ.65 ) GO TO 41
        AB(I1,J1)=X
41      IF (I2.EQ.65) GO TO 40
        AB(I2,J1)=X
40      IF( J2.EQ.65 ) GO TO 20
        IF( I1.EQ.65 ) GO TO 42
        AB(I1,J2)=X
42      IF(I2.EQ.65) GO TO 20
        AB(I2,J2)=X
20      CONTINUE
        RETURN
        END


        SUBROUTINE TRANSFOR(INDICE)
        COMMON /BUFFER/ A
        COMPLEX A(64,64),AA(4096),WORK(64)
        EQUIVALENCE (A,AA)
        REAL AAA(64,64),RA(4096),BB(64,64)
        EQUIVALENCE (A,AA,AAA,RA),(AA(2049),BB)
        IF (INDICE.EQ.1) THEN
C       TRANSFORMADA DIRECTA DE FFT.
        PRINT *,' Transformada directa de fourier ( FFT )'
                          SIGNO=+1.0
                          K=1
                          DO 100 I=1,64
                          K=3-K
                          DO 100 J=K,64,2
100                       A(I,J)=-A(I,J)
                          DO 110 I=1,64
                          CALL FASTFFT(A(1,I),SIGNO)
110                       CONTINUE
                          DO 120 J=1,64
                          DO 125 I=1,64
                          WORK(I)=A(J,I)
125                       CONTINUE
                          CALL FASTFFT(WORK,SIGNO)
                          DO 126 I=1,64
                          A(J,I)=WORK(I)
126                       CONTINUE
120                       CONTINUE
                          K=1
                          DO 130 I=1,64
                          K=3-K
                          DO 130 J=K,64,2
                          A(I,J)=-A(I,J)
130                       CONTINUE
                          RETURN
                          ELSEIF (INDICE.EQ.2) THEN
C       TRANSFORMADA INVERSA DE FOURIER.
                          SIGNO=-1.0
                          K=1
                          DO 200 I=1,64
                          K=3-K
                          DO 200 J=2,64,2
200                       A(I,J)=-A(I,J)
                          DO 210 I=1,64
```

```
                        CALL FASTFFT(A(1,I),SIGNO)
 210                    CONTINUE
                        DO 220 J=1,64
                        DO 225 I=1,64
                        WORK(I)=A(J,I)
 225                    CONTINUE
                        CALL FASTFFT(WORK,SIGNO)
                        DO 226 I=1,64
                        A(J,I)=WORK(I)
 226                    CONTINUE
 220                    CONTINUE
                        K=1
                        DO 230 I=1,64
                        K=3-K
                        DO 230 J=K,64,2
                        A(I,J)=-A(I,J)
 230                    CONTINUE
                        RETURN
                        ENDIF
                        END


       SUBROUTINE FASTFFT(A,SIGNO)

       COMPLEX A(64),U,DOBLEW,TE
       DATA NM1,NV2,N,M,PI/63,32,64,6,3.141592653589793/
       J=1
       DO 100 I=1,NM1
       IF (I.GE.J) GO TO 101
       TE=A(J)
       A(J)=A(I)
       A(I)=TE
 101   K=NV2
 102   IF(K.GE.J) GO TO 103
       J=J-K
       K=K/2
       GO TO 102
 103   J=J+K
 100   CONTINUE
       DO 110 L=1,M
       LE=2**L
       LE1=LE/2
       U=(1.0,0.0)
       ANGULO=PI/FLOAT(LE1)
       DOBLEW=CMPLX(COS(ANGULO),-SIGNO*SIN(ANGULO))
       DO 120 J=1,LE1
       DO 115 I=J,N,LE
       IP=I+LE1
       TE=A(IP)*U
       A(IP)=A(I)-TE
       A(I)=A(I)+TE
 115   CONTINUE
       U=U*DOBLEW
 120   CONTINUE
 110   CONTINUE
       DO 130 I=1,64
       A(I)=A(I)/8.0
 130   CONTINUE
       END
```

```
      SUBROUTINE INOUT(OPCION,N,BUFFER,CANAL)
      COMPLEX BUFFER(N)
      INTEGER OPCION,CANAL

      LARGO=64*N
      IF (OPCION.EQ.1) THEN
                       DO 100 I=1,LARGO,64
                       K1=I
                       K2=I+63
                       READ(CANAL) (BUFFER(K),K=K1,K2)
100                    CONTINUE
                       RETURN
                       ELSEIF (OPCION.EQ.2) THEN
                       DO 200 I=1,LARGO,64
                       K1=I
                       K2=I+63
                       WRITE(CANAL) (BUFFER(K), K=K1,K2)
200                    CONTINUE
                       RETURN
                       ENDIF

      PRINT *,' ERROR: No existe la opcion que me',
     *' indico en uno de los parametros'
      END
```

```
  NIVELES MIN, MAX :  0.0000000E+00   1.000000
-32 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-31 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-30 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-29 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-28 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-27
-26
-25
-24
-23
-22 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-21 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-20 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-19 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-18 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-17
-16
-15
-14
-13
-12 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-11 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
-10 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 -9 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 -8 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 -7
 -6
 -5
 -4
 -3
 -2 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 -1 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
  0 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
  1 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
  2 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
  3
  4
  5
  6
  7
  8 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
  9 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 10 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 11 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 12 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 13
 14
 15
 16
 17
 18 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 19 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 20 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 21 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 22 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 23
 24
 25
 26
 27
 28 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 29 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 30 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
 31 ££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££££
```

Fig. 1. Test de patrones horizontales.

```
 NIVELES MIN, MAX :  0.0000000E+00   1156.000
-32
-31
-30
-29
-28
-27
-26
-25
-24
-23
-22
-21
-20
-19
-18
-17
-16
-15
-14
-13
-12
-11
-10
 -9
 -8
 -7
 -6                                  +
 -5
 -4
 -3
 -2
 -1
  0                                  £
  1
  2
  3
  4
  5
  6                                  +
  7
  8
  9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
```

Fig. 2. Módulo de la transformada de Fourier de la Fig. 1.

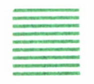
```
NIVELES MIN, MAX :  0.0000000E+00   1.000000
 -32
 -31
 -30
 -29
 -28
 -27
 -26
 -25
 -24
 -23
 -22
 -21
 -20
 -19
 -18
 -17
 -16
 -15
 -14
 -13
 -12
 -11
 -10
  -9
  -8
  -7
  -6
  -5                                     ££££££££££
  -4                                     ££££££££££
  -3                                     ££££££££££
  -2                                     ££££££££££
  -1                                     ££££££££££
   0                                     ££££££££££
   1                                     ££££££££££
   2                                     ££££££££££
   3                                     ££££££££££
   4                                     ££££££££££
   5
   6
   7
   8
   9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
```

Fig. 3. Test rectangular.

```
 NIVELES MIN, MAX :  0.0000000E+00   2.441406
-32
-31
-30
-29
-28
-27
-26
-25
-24
-23
-22
-21
-20
-19
-18
-17
-16
-15
-14
-13
-12
-11
-10
 -9
 -8
 -7
 -6
 -5
 -4                               +++++
 -3                              +%***%+
 -2                             +%$&&&$%+
 -1                             +*&§£§&*+
  0                             +*&£££&*+
  1                             +*&§£§&*+
  2                             +%$&&&$%+
  3                              +%***%+
  4                               +++++
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
```

Fig. 4. Módulo de la transformada de Fourier de la Fig. 3.

APENDICE 3



BIBLIOGRAFIA

ANDREWS, H. C.: *Computer Techniques in Image Processing*. Academic Press, 1970.

ANDREWS, H. C.: «Tutorial and Selected Papers in Digital Image Processing», *IEEE Computer Society*, 1978.

ANDREWS, H. C.: *Digital Image Restoration*. Prentice Hall, 1977.

BALLARD, D. H., y BROWN, C. M.: *Computer Vision*. Prentice Hall, 1982.

BOW, S. T.: *Pattern Recognition*. Marcel Dekker, 1984.

BRODATZ, P.: *Textures*. Dover, 1966.

CASASENT, D.: *Optical Data Processing*. Springer-Verlag, 1978.

CASTLEMAN, K. R.: *Digital Image Processing*. Prentice Hall, 1979.

DEVIJVER, P. A., y cols.: *Pattern Recognition: A Statistical Approach*. Prentice Hall, 1982.

DODD, G. G., y ROSSOL, L.: *Computer Vision and Sensor-Based Robots*. Plenum Press, 1979.

DUDA, R. O., y HART, P. E.: *Pattern Classification and Scene Analysis*. Wiley, 1973.

GONZÁLEZ, R. C., y WINTZ, P.: *Digital Image Processing*. Addison Wesley, 1977.

DUFF, M. J. B., y LEVIALDI, S.: *Languages and Architectures for Image Processing*. Academic Press, 1981.

EKSTROM, M. P. (ed.): *Digital Image Processing Techniques*. Academic Press, 1984.

FU, K. S.: *Syntactic Methods in Pattern Recognition*. Academic Press, 1974.

FU, K. S. (ed.): *Syntactic Pattern Recognition: Applications*. Springer-Verlag, 1977.

FU, K. S.: *Applications of Pattern Recognition*. CRC Press, 1982.

FUKUNAGA, K.: *Introduction to Statistical Pattern Recognition*. Academic Press, 1972.

GOODMAN, J. W.: *Introduction to Fourier Optics*. McGraw Hill, 1968.

GREEN, W. B.: *Digital Image Processing: A System Approach*. Van Nostrand, 1963.

HALL, E. L.: *Computer Image Processing and Recognition*. Academic Press, 1979.
HORN, B. K. P.: *Robot Vision*. MIT Press, 1986.
HANSON, A. R., y RISEMAN, E. M. (eds.): *Computer Vision System*. Academic Press, 1978.
HARALICK, R. M.: «Pictorial Data Analysis», *NATO ASI Series*. Springer-Verlag, 1983.
HUANG, T. S. (ed.): *Picture Processing and Digital Filtering*. Springer-Verlag, 1975.
HUANG, T. S., y TRETIAK, O. J. (eds.): *Picture Bandwidth Compression*. Gordon and Breach, 1972.
KLINGER, A., y cols.: *Data Structures, Computer Graphics and Pattern Recognition*. Academic Press, 1977.
LEVINE, M. D.: *Vision in Man and Machine*. McGraw Hill, 1985.
LINDSAY, P. H., y NORMAN, D. A.: *Human Information Processing*. Academic Press, 1972. Existe traducción al castellano: *Procesamiento de la Información Humana*. Tecnos, 1975.
LINTZ, J., y SIMONET (eds.): *Remote Sensing Environment*. Addison Wesley, 1976.
MARR, D.: *Vision*. Freeman, 1982. Existe traducción en castellano: *La Visión*. Alianza Editorial, 1986.
MICLET, L.: *Méthodes Structurelles pour la Reconnaissance des Formes*. Eyrolles, 1984.
NAGAO, M., y MATSUYAMA, T.: *A Structural Analysis of Complex Aerial Photographs*. Plenum Press, 1980.
NILSSON, N. J.: *Problem-Solving Methods in Artificial Intelligence*. McGraw Hill, 1971.
NUDELMAN, S., y cols. (eds.): «Imaging for Medicine», vol. 1, *Nuclear Medicine, Ultrasonics and Thermography*. Plenum Press, 1980.
OKOSHI, T.: *Three Dimensional Imaging Techniques*. Academic Press, 1976.
PATRICK, E.: *Fundamentals of Pattern Recognition*. Prentice Hall, 1979.
PAVLIDIS, T.: *Structural Pattern Recognition*. Springer-Verlag, 1977.
PAVLIDIS, T.: «Algorithms for Graphics and Image Processing», *Computer Science Press*, 1982.
PRATT, W. K.: *Digital Image Processing*. John Wiley, 1978.
PRESTON, J. K, y cols. (eds.): *Medical Imaging Techniques. A Comparison*. Plenum Press, 1979.
ROSENFELD, A.: *Digital Picture Processing*, 2.ª ed. Academic Press, 1982.
SERRA, J.: *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
TOU, J. T. y GONZÁLEZ, R. C.: *Pattern Recognition Principles*. Addison Wesley, 1974.
YOUNG y FU (eds.): *Handbook on Pattern Recognition and Image Processing*. Academic Press, 1986.

APENDICE 4



GLOSARIO

Imagen. Representación espacial de un objeto, escena u otra imagen. El nivel de gris I(x,y) correspondiente a la coordenada espacial (x,y) es, en general, proporcional a la energía radiante.

Nivel de gris. Corresponde al número o valor asignado a la posición (x,y) de una imagen. Este número es proporcional a la reflectancia o transmitancia de una pequeña área denominada pixel.

Intensidad radiante. De un punto o un objeto es una medida de la potencia radiante por estereorradián radiado o reflejado por un objeto. La intensidad radiante es una función de la naturaleza del objeto, ángulo de observación, longitud de onda espectral y ancho de banda.

Reflectancia o coeficiente de reflexión. Es el cociente entre la energía por unidad de tiempo y por unidad de área (densidad de potencia radiante) reflejada por un objeto y la energía por unidad de tiempo y por unidad de área incidente en el objeto.

Transmitancia o coeficiente de transmitancia. Es el cociente entre la energía por unidad de tiempo y unidad de área (densidad de potencia radiante) transmitida por un objeto y la energía por unidad de tiempo y unidad de área incidente en el objeto.

Contraste. El contraste de un punto frente a un fondo puede medirse por: 1) su razón de contraste, que es el cociente entre el valor mayor de la transmitancia del objeto o del fondo y el valor menor de la transmitancia del objeto o del fondo, y 2) *diferencia de contraste,* que es la diferencia entre el valor mayor de la transmitancia del objeto o del fondo y el valor menor de la transmitancia del objeto o del fondo.

Densidad. La densidad en un punto (x,y) de una fotografía es una medida de la capacidad de absorción de la luz de la plata o del tinte depositado en cada posición.

Densitómetro. Dispositivo utilizado para medir el valor medio de la densidad de la imagen correspondiente a una pequeña área de una fotografía o de una transparencia. Cuando el área es menor que unos cente-

nares de micras cuadradas, el dispositivo recibe el nombre de microdensitómetro.

Resolución. El término resolución hace referencia a cómo un sistema, proceso, componente o material puede reproducir un objeto aislado o discriminar entre objetos próximos. El límite de resolución o resolución espacial viene dada por la dimensión más pequeña del objeto que puede ser discriminada.

Poder de resolución. El poder de resolución de un sistema de tratamiento de imágenes es una medida de su capacidad de discriminación entre objetos próximos. El poder de resolución se mide en pares de líneas por milímetro, es decir, el máximo número de líneas y espacios por milímetro que pueden reconocerse.

Imagen de vídeo. Es una imagen que presenta un formato capaz de ser presentada en un tubo de rayos catódicos.

Cuantificación. Proceso mediante el cual cada nivel de gris de una imagen, fotografía, etc., se le asigna un valor entre un conjunto finito de niveles de gris.

Filtro espacial. Transformación de una imagen utilizada para reducir el ruido o mejorar algunas características de la imagen.

El procesado de imágenes es un campo de reciente y rápido desarrollo, con importantes aplicaciones en áreas tan diversas como la mejora de imágenes biomédicas, robóticas, teledetección y otras aplicaciones industriales y militares. Se presentan los principios básicos, los sistemas y las técnicas de procesado más usuales.